



ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Αχιλλέας Ματσούκας 2114025

achilleasmatsoukas@gmail.com

Κατανεμημένη Επεξεργασία Ροών Δεδομένων στο Διαδίκτυο των Πραγμάτων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

.....Ονοματεπώνυμο Υπεύθυνου.....

.....Βαθμίδα.....

ΣΥΝΕΠΙΒΛΕΠΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Κατανεμημένη Επεξεργασία Ροών Δεδομένων στο Διαδίκτυο των Πραγμάτων

Αχιλλέας Ματσούκας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Δρ Κων/νος Κολομβάτσος

Επίκουρος Καθηγητής Πανεπιστημίου Θεσσαλίας

Λαμία έτος



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Distributed Computing of Data Streams in Internet of Things

Achilleas Matsoukas

FINAL THESIS

ADVISOR

Dr Konstantinos Kolomvatsos

Lamia year

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΙΟΤ Θερμοκήπιο



Περιεχόμενα

1. Διαδίκτυο των Πραγμάτων (IOT)

<i>1.1 Ιστορική Αναδρομή του Διαδικτύου των πραγμάτων</i>	<i>15</i>
<i>1.2 Ορισμός Διαδικτύου των Πραγμάτων</i>	<i>15</i>
<i>1.3 Χαρακτηριστικά Διαδικτύου των Πραγμάτων</i>	<i>16</i>
<i>1.4 Κύρια Μέρη του Διαδικτύου των πραγμάτων</i>	<i>17</i>
<i>1.5 Παράδειγμα IOT</i>	<i>18</i>

2. Αισθητήρες

<i>2.1 Ορισμός Αισθητήρων</i>	<i>19</i>
<i>2.2 Τύποι Αισθητήρων</i>	<i>19</i>

3. Μηχανική Μάθηση

<i>3.1 Εισαγωγή Μηχανικής Μάθησης</i>	<i>21</i>
<i>3.2 Ορισμός Μηχανικής Μάθησης</i>	<i>22</i>
<i>3.3 Μοντέλα & Πρότυπα Μηχανικής Μάθησης</i>	<i>23</i>
<i>3.3.1 Ορισμός Μοντέλων και Προτύπων</i>	<i>26</i>
<i>3.3.2 Κατηγορίες Μοντέλων και Προτύπων</i>	<i>27</i>

4. Διαχείριση Δεδομένων

<i>4.1 Ορισμός Διαχείρισης Δεδομένων</i>	<i>28</i>
<i>4.2 Διαχείριση Big Data</i>	<i>30</i>
<i>4.3 Μηχανική Μάθηση & Δεδομένα</i>	<i>31</i>

5. Συστήματα Διαχείρισης Γνώσης

5.1 Αντιπροσωπεύοντας τη γνώση	30
5.2 Βασικά συστήματα διαχείρισης	32
 6. Προτεινόμενο Σύστημα	
6.1 Περίληψη Συστήματος	34
6.2 Υλικό / Hardware	34
6.3 Λογισμικό / Software	43
6.3.1 Αισθητήρες	43
6.3.2 Machine Learning LSTM	45
6.3.3 Rule Based System (Durable)	49
6.3.4 Υλικές Λειτουργίες	51
6.3.5 Αυτόματο Πότισμα	53
 7. Συμπεράσματα και Μελλοντικές Προεκτάσεις	
7.1 Συμπεράσματα	56
7.2 Μελλοντικές Προεκτάσεις	57

Σε μια εποχή όπου η τεχνολογία κάθε μέρα αναπτύσσεται με ρυθμό μεγαλύτερο από ποτέ, δημιουργούνται ανάγκες που δεν είχαμε ποτέ σκεφτεί. Τα έξυπνα συστήματα έχουν υιοθετηθεί από τη βιομηχανία αλλά και στην καθημερινότητα μας. Οι ανάγκες για νέες , βέλτιστες και οικονομικές καινοτομίες είναι ένα βασικό κριτήριο της οικονομίας του σήμερα. Το Διαδίκτυο των Πραγμάτων είναι μια από τις βασικότερες τεχνολογίες για την δημιουργία αυτών των καινοτομιών καθώς ενοποιεί όλες τις δραστηριότητες σε ένα καθολικό κατανεμημένο σύστημα το οποίο , μπορεί να έχει τον έλεγχο πάνω σε ένα σύστημα και να το προγραμματίζει όπως ένας άνθρωπος αλλά γρηγορότερα και αποδοτικότερα. Με την χρήση της Μηχανικής Μάθησης και τον καθορισμό κανόνων μπορούμε να “εκπαιδεύουμε” ένα σύστημα να κατανοεί και να ελέγχει ένα σύνολο εργαλείων και προγραμμάτων όπως ακριβώς ένας άνθρωπος , έχοντας το πλεονέκτημα της μελέτης των δεδομένων για πρόβλεψη μελλοντικών αλλαγών ώστε να αποτρέπονται σενάρια τα οποία θα ήταν καταστροφικά ή απλά μη αποδοτικά.

Στην εργασία αυτή χρησιμοποιώντας όλα αυτές τις τεχνολογίες, κατασκευάσαμε ένα αυτόματο θερμοκήπιο το οποίο χωρίς την ανθρωπινή επιβλέψη μπορεί και παίρνει αποφάσεις ανάλογα με τις περιβαλλοντικές συνθήκες και βάσει κάποιων προκαθορισμένων κανόνων , να παίρνει αποφάσεις ακριβώς όπως θα τις έπαιρνε ένας γεωπόνος, αυτοματοποιώντας τη διαδικασία της καλλιέργειας φυτών με ίδια και καλύτερα αποτελέσματα από το συμβατικό τρόπο όπου θα χρειαζόταν να υπάρχει καθημερινή επίβλεψη.

Abstract

We are living in an era in which technology keeps growing and upgrading with a rate never seen before, creating and inventing needs that humans never depicted before. Intelligent systems have been embraced by the Industry and our daily life. The need for new, innovating and quality products is a main focus for the growth of the economy in today's standard. Internet of Things is one of the primary technology, in order to succeed this innovation, using a unanimous distributed system It can embed all the actions and control over a system to achieve what a human can do , but faster and better. Using Machine Leering and a Rule Based system, we can set up a system that can be trained to have control over a structure and knowledge over it, with the advantage it can predict events that must be prevented or mitigated, using the data streams that the system produces.

Using the knowledge and the technologies mentioned above, in this thesis we constructed and developed an automated GreenHouse, which can function without the human need, using the data streams and the rules it can take actions and make decisions using the environmental data, achieving what an expert in the agricultural domain could have. Resulting in the automation of agriculture producing same level or better outcomes, without the need of constant supervision.

1.Διαδίκτυο των Πραγμάτων

1.1 Ιστορική Αναδρομή του Διαδικτύου των πραγμάτων

Η εμφάνιση και ο ορισμός του Internet of things (IOT), <<Διαδίκτυο των Πραγμάτων >> όπως θα μπορούσε να ονομαστεί στα Ελληνικά, χρησιμοποιήθηκε για πρώτη φορά το 1999 από τον επιχειρηματία Kevin Ashton.

Μέσα από την επιστημονική του έρευνα παρουσίασε τη σύνδεση αντικειμένων με το internet χάρη στη χρήση ραδιοσυχνοτήτων.

Καθώς τα χρόνια κυλούν, παρατηρούμε πως όλο και περισσότερες συσκευές αποκτούν αισθητήρες και μας παρέχουν τη δυνατότητα να τις συνδέουμε και να τις ελέγχουμε μέσω του κινητού μας τηλεφώνου ή υπολογιστή. Αυτό γιατί οι αισθητήρες μας δίνουν τη δυνατότητα στις συσκευές να συνδέονται στο παγκόσμιο ιστό. Όλο αυτό μας δίνει το αποτέλεσμα της μεταξύ τους επικοινωνίας, αλλά και της επικοινωνίας με τον άνθρωπο.

Τα πρώτα μοντέλα που θα μπορούσαν να περιγράψουν τον ορισμό του Internet of things είναι τα έξυπνα σπίτια. Δηλαδή να σου δίνεται η δυνατότητα να ελέγχεις το σπίτι σου από απόσταση. Μέσω για παράδειγμα του κινητού σου τηλεφώνου να μπορείς να ελέγξεις από το γενικό του σπιτιού σου μέχρι τις πιο μεμονωμένες συσκευές, βλέπε πλυντήριο ρούχων ή καφετιέρα, σύστημα κλειδώματος μέχρι σύστημα ψεκασμού γκαζόν. Δημιουργήθηκε η δυνατότητα να υπάρξουν ξεχωριστά προφίλ λειτουργιάς των συσκευών.

1.2 Ορισμός Διαδικτύου των Πραγμάτων

Το Διαδίκτυο των πραγμάτων ή Ίντερνέτ των πραγμάτων (αγγλικά: Internet of things) αποτελεί το δίκτυο επικοινωνίας πληθώρας συσκευών, οικιακών συσκευών, αυτοκινήτων καθώς και κάθε αντικειμένου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων. Απλούστερα, η φιλοσοφία του IoT είναι η σύνδεση όλων των ηλεκτρονικών συσκευών μεταξύ τους (τοπικό δίκτυο) ή με δυνατότητα σύνδεσης στο διαδίκτυο (παγκόσμιο ιστό).



Εικόνα 1 - Διαδίκτυο των πραγμάτων

Η έννοια "Things" (πράγματα) δεν είναι αυστηρά συνδεδεμένη με ορισμένα προϊόντα. Αναφέρεται σε μία ευρεία ποικιλία συσκευών εντελώς διαφορετικά μεταξύ τους, όπως για παράδειγμα αυτοκίνητα με ενσωματωμένους αισθητήρες, κάμερες, κλιματιστικά, φώτα, συστήματα ασφαλείας, smartwatches ακόμα και αυτοκίνητα των οποίων οι περίπλοκοι αισθητήρες εντοπίζουν αντικείμενα στην πορεία τους. Είναι μερικά από τα πολλά προϊόντα τεχνολογίας. Βασικό χαρακτηριστικό όλων είναι η σύνδεση μεταξύ τους με απώτερο σκοπό τη δυνατότητα του χρήστη να τα ελέγχει από έναν υπολογιστή ή κινητό. Ο όρος Internet of Things αποδόθηκε την δεκαετία του 1990 από τον Kevin Ashton[8].

Το Διαδίκτυο των Πραγμάτων (Internet of Things) είναι μία από τις τρεις κορυφαίες τεχνολογικές εξελίξεις της επόμενης δεκαετίας (μαζί με το mobile Internet και την αυτοματοποίηση της γνώσης εργασίας) και αποτελεί το επόμενο μεγάλο βήμα στον χώρο της τεχνολογίας. Ο Ashton, ο οποίος είναι ένας από τους ιδρυτές του Auto-ID center στο MIT, ήταν μέρος μιας ομάδας που ανακάλυψε τον τρόπο να συνδέσει τα αντικείμενα με το Διαδίκτυο μέσω μιας ετικέτας RFID. [8]

1.3 Χαρακτηριστικά Διαδικτύου των Πραγμάτων

1. Ευφυΐα – Ο συνδυασμός του υλικού και του λογισμικού σε συνεργασία με πολύπλοκους αλγορίθμους, καθιστά το Διαδίκτυο των πραγμάτων ένα έξυπνο σύστημα το οποίο είναι σε θέση να ανταπεξέρθει και να ενεργήσει ανάλογα με την κάθε κατάσταση.

2. Δυναμική φύση – Έχοντας τη δυνατότητα για άμεση επαφή με όλο το δίκτυο και το περιβάλλον του και ανταλλάζοντας πληροφορίες μεταξύ των φυσικών κόμβων μπορεί να αλλάξει τα αποτελέσματα του ανάλογα με της πληροφορίες σε πραγματικό χρόνο.

3. Επεκτασιμότητα – Σε μια τεχνολογία όπου κάθε κόμβος , συνεφέρει στο τελικό αποτέλεσμα και στη επεξεργασία των πληροφοριών , είναι εύκολο να γίνει επέκταση χωρίς να υπάρχει σημαντική επιβάρυνση στο σύστημα.

4. Αισθητήρες – Είναι ένα από τα πιο βασικά κομμάτια του Διαδικτύου των πραγμάτων διότι , με αυτούς επιτυγχάνεται η άμεση επικοινωνία με το φυσικό περιβάλλον παράγοντας της πληροφορίες που θέλουμε.

5. Διαφορετικότητα – Ανεξαρτήτως της αρχιτεκτονικής η του λογισμικού που έχει ένας κόμβος , το Διαδίκτυο των πραγμάτων είναι σε θέση να υποστήριξη την επικοινωνία ανάμεσα σε διαφορετικά δίκτυα και κόμβους. Αυτό έχει ως απώτερο σκοπό την επεκτασιμότητα και την συμβατότητα ανάμεσα σε όλα τα συστήματα.

1.4 Κύρια Μέρη του Διαδικτύου των πραγμάτων

Τα τρία κύρια μέρη ενός IoT είναι:

1)Οι συσκευές που συλλέγουν οπουδήποτε και οποιαδήποτε στιγμή χρησιμοποιώντας RFID τεχνολογία, αισθητήρες και κώδικα. Όλες αυτές οι πληροφορίες που συλλέγονται μπορεί να ποικίλουν σε πολυπλοκότητα , από ένα απλό αισθητήρα θερμοκρασίας έως ένα σύστημα μετάδοσης ζωντανής εικόνας. Για παράδειγμα τα προσωπικά κινητά μας εξοπλισμένα με αισθητήρες όπως είναι τα επιταχυνόμετρα, γυροσκόπια, κ.λπ. Τα δεδομένα που συλλέγονται από αυτά τα τερματικά σημεία μπορούν να χρησιμοποιηθούν σε διάφορους τομείς, όπως είναι η αναγνώριση ανθρώπινης κίνησης.

2)Τα δίκτυα επικοινωνίας που συνδέουν τις συσκευές αυτές. Σε ένα τυπικό σύστημα Διαδίκτυο των Πραγμάτων , οι αισθητήρες είναι πάντα συνδεδεμένοι στα υπολογιστικά συστήματα μέσω επιπέδων δικτύων και επιπέδων συνδεσιμότητας. Οι συσκευές είναι πάντα συνδεδεμένες μεταξύ τους εντός του επιπέδου συνδεσιμότητας. Ανάλογα με το μέγεθος της εφαρμογής μας οι συσκευές μας μπορούν να συνδεθούν μέσω LAN, MAN, WAN ή και μέσω τηλεφωνικών δικτύων όπως είναι το 4G,5G.

3)Τα υπολογιστικά συστήματα και οι εφαρμογές που επεξεργάζονται όσα δεδομένα ρέουν από και προς τις συσκευές αυτές. Μόλις το σύστημα έχει στην κατοχή του τα δεδομένα που σύλλεξε από τους αισθητήρες , το λογισμικό έχει το ρόλο για την επεξεργασία τους. Αυτό μπορεί να είναι ένας απλός έλεγχος όπου το λογισμικό βλέπει αν η θερμοκρασία

που διαβάστηκε από ένα κλιματιστικό , είναι εντως αποδεχτών τιμών , μέχρι και πολύπλοκους αλγορίθμους που αναγνωρίζουν αντικείμενα μέσω επεξεργασίας βίντεο.

1.5 Παράδειγμα IOT

1. Σε κτηνοτρόφο

Μπορεί εύκολα με τη βοήθεια κτηνιάτρου να τοποθετήσει μια συσκευή ιοτ σε κάθε ζώο που υπάρχει στην ιδιοκτησία του, η οποία συσκευή περιχεί σύστημα εντοπισμού με την χρήση GPS , σύστημα καταγραφής των ζωτικών στοιχείων του ζώου (π.χ. κτύπους καρδιάς) και μπορεί ανά πασα στιγμή να εντοπίζει που είναι το κοπάδι του, καθώς και την συμπεριφορά των ζώων. Χωρίς την ανθρωπινή παρουσία γίνεται βατή η βόσκηση σε ελεγχόμενο χώρο (γεωφράχτη) , καθώς και ο έλεγχος για εξωτερικούς παράγοντες, όπως θα ήταν για παράδειγμα ένας λύκος για ένα κοπάδι προβατα.[2]

2. Σε ιατρική συσκευή

Πάλι με το ίδιο σκεπτικό όπως και στους κτηνοτρόφους βλέπουμε πως ο γιατρός μας μπορεί ανά πασα στιγμή να ελέγχει τις μετρήσεις του ασθενή. άμεση και με ακρίβεια καταγραφή τιμών μετρήσεων των ασθενών σε πραγματικό χρόνο. Χρησιμοποιώντας αισθητήρες για παλμούς, θερμοκρασία, αναπνοή, μπορούμε να δημιουργήσουμε μοντέλα ώστε να προβλέψουμε αν ένας ασθενής πρόκειται να πάθει κάτι. Ένα παράδειγμα είναι, ότι με την χρήση αυτού του συστήματος μπορούμε να εγκαταστήσουμε σε ανθρώπους που έχουν κάποια νόσο όπως είναι κρίση πανικού, καρδιακά επεισόδια, εγκεφαλικά κ.λπ. και με την χρήση δεδομένων να προλάβουμε αυτές τις καταστάσεις βρίσκοντας της μεταβλητές που τις προκαλούν πριν καν γίνουν.[2]

3. στη ιδέα των έξυπνων πόλεων

Πιο συγκεκριμένα σε στύλους φωτισμού ή έξυπνοι μετρητές υπηρεσιών ωφελείας, οι ΙοΤ συσκευές προσαρμόζουν τη ισχύ σύμφωνα με το φυσικό φως ή την χαμηλώνουν με την χρήση αισθητήρων κίνησης όταν δεν ανιχνεύουν ανθρώπους ή με την χρήση αισθητήρων φωτός για να ανοίγουν μονό όταν δεν υπάρχει αρκετό φυσικό φως. Μπορούν επίσης με την χρήση αισθητήρων ανιχνεύεις μετάλλων , να γίνει ένα σύστημα στάθμευσης το οποίο θα γνωρίζει σε πραγματικό χρόνο ποιες θέσεις είναι ελεύθερες και θα ενημερώνει τους οδηγούς. Οι μετρητές ΙοΤ ενημερώνουν για τα χαρακτηριστικά της κατανάλωσης (ποσότητα, ώρες χρήσης κλπ). Έτσι κερδίζουμε χαμηλότερη κατανάλωση ρεύματος, ασφάλεια και λιγότερους ρύπους.

Τους αισθητήρες μπορούμε να τους βρούμε σε πολλά σχήματα και μεγέθη. Ανάλογα με τις ανάγκες μας και τους διαθέσιμους πόρους. Μερικοί από αυτούς είναι:

1. Αισθητήρας Θερμοκρασίας : Ο οποίος μετράει την ποσότητα της θερμικής ενεργείας από μια πηγή, και του επιτρέπεται να εντοπίζει και να μετατρέπει αυτές τις αλλαγές σε πληροφορία.
2. Αισθητήρας Υγρασίας : Αυτός ο τύπος αισθητήρα μετρά την ποσότητα υδρατμών που υπάρχουν στον αέρα ή σε άλλα αέρια.
3. Αισθητήρας Πίεσης : Ένας αισθητήρας πίεσης εντοπίζει τις αλλαγές σε ένα αέριο ή μια υγρή ουσία.
4. Αισθητήρας Κίνησης : Χρησιμοποιείται για την εντόπιση χωρίς επαφή , για ένα αντικείμενο που είναι κοντά στον αισθητήρα. Με τη χρήση ηλεκτρομαγνητικών κυμάτων ή υπερύθρων ακτίνων.
5. Αισθητήρας Στάθμης : Χρησιμοποιείται για την ανίχνευση της στάθμης μιας ουσίας συμπεριλαμβάνοντας τα υγρά, τη σκόνη και ότι υπάρχει σε κοκκώδη μορφή.
6. Αισθητήρας Επιτάχυνσης : Ανιχνεύει την αλλαγή της ταχύτητας ενός αντικείμενου σε σχέση με το χρόνο. Μπορούν επίσης και να ανιχνεύουν αλλαγές στη βαρύτητα.
7. Γυροσκόπιο : Το γυροσκόπιο μπορεί να μετρήσει τις αλλαγές στην γωνιακή ταχύτητα ή αλλιώς την ταχύτητα και την περιστροφή ενός αντικείμενου γύρω από ένα άξονα.
8. Οπτικός : Μπορεί να μετατρέψει της ακτίνες του φωτός σε ηλεκτρικά σήματα.

3.1 Εισαγωγή Μηχανικής Μάθησης

Μια από τις θεμελιώδεις ιδιότητες / ικανότητες του ανθρώπινου είδους είναι η μάθηση. Έτσι δημιουργείται η ιδέα του πως θα μπορούσε ο επιστημονικός τομέας να δημιουργήσει υπολογιστικά συστήματα τα οποία θα μπορούσαν να μάθουν. Πιο συγκεκριμένα ο τρόπος να εκπαιδεύσουμε έναν υπολογιστή να μπορεί να κατανοεί χωρίς να εμείς να ορίζουμε ρητά κάποιο πρόγραμμα / κανόνες.

Η μηχανική της εκμάθησης μας βοηθάει να αξιοποιηθεί η προηγούμενη γνώση και εμπειρία και να αποδοθεί στη δημιουργία μηχανών ικανών να μαθαίνουν και να εξελίσσονται .

3.2 Ορισμός Μηχανικής Μάθησης



Εικόνα 3 – Μηχανική μάθηση

Ο Tom M. Mitchell πρότεινε έναν πιο επίσημο ορισμό που χρησιμοποιείται ευρέως: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P , αν η επίδοσή του σε εργασίες της κλάσης T , όπως αποτιμάται από το μέτρο P , βελτιώνεται με την εμπειρία E ». Αυτός ο ορισμός είναι σημαντικός για τον καθορισμό της μηχανικής μάθησης σε βασικό λειτουργικό πλαίσιο παρά με γνωστικούς όρους, ακολουθώντας έτσι την πρόταση του Alan Turing στην εργασία του «Υπολογιστικές μηχανές και Νοημοσύνη», ότι το ερώτημα αν μπορούν οι μηχανές να σκεφτούν, μπορεί να αντικατασταθεί με το ερώτημα αν μπορούν οι μηχανές να κάνουν αυτό που εμείς (ως σκεπτόμενες οντότητες) μπορούμε να κάνουμε. [6]

3.3 Μοντέλα & Πρότυπα Μηχανικής Μάθησης

3.3.1 Ορισμός Μοντέλων και Προτύπων

Μοντέλα και πρότυπα είναι το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μια συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου.

Στον ανθρώπινο νου είναι πιο εύκολο να συλλέξει δεδομένα και να τα επεξεργαστεί, να τα κρίνει και να τα κατανέμει αντίστοιχα. Πως όμως αυτό εξηγείται με τη μηχανική μάθησης;

Εδώ έχουμε την ικανότητα ενός υπολογιστικού συστήματος να δημιουργεί μοντέλα ή πρότυπα από ένα σύνολο δεδομένων. Μετά την ολοκλήρωση της εκπαίδευσης του μοντέλου το σύστημα είναι σε θέση να κάνει προβλέψεις και να επεξεργαστεί δεδομένα τα οποία δεν έχει ξαναδεί ποτέ. Για παράδειγμα μπορούμε να εκπαιδεύουμε ένα μοντέλο να αναγνωρίζει πρόσωπα και εκφράσεις με την τροφοδότηση κάποιων εικόνων με πρόσωπα, και μετά με την χρήση αυτού του μοντέλου μπορούμε να αναγνωρίζουμε οποιοδήποτε νέο πρόσωπο.

Τα συγκεκριμένα μοντέλα ή πρότυπα μπορούν να κάνουν προβλέψεις ή να μας δώσουν αποφάσεις. Μια πανομοιότυπη σκέψη με την υπολογιστική στατιστική.

3.3.2 Κατηγορίες Μοντέλων και Προτύπων

Οι εργασίες μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις μεγάλες κατηγορίες, ανάλογα με τη φύση του εκπαιδευτικού «σήματος» ή την «ανατροφοδότηση» που είναι διαθέσιμα σε ένα σύστημα εκμάθησης.

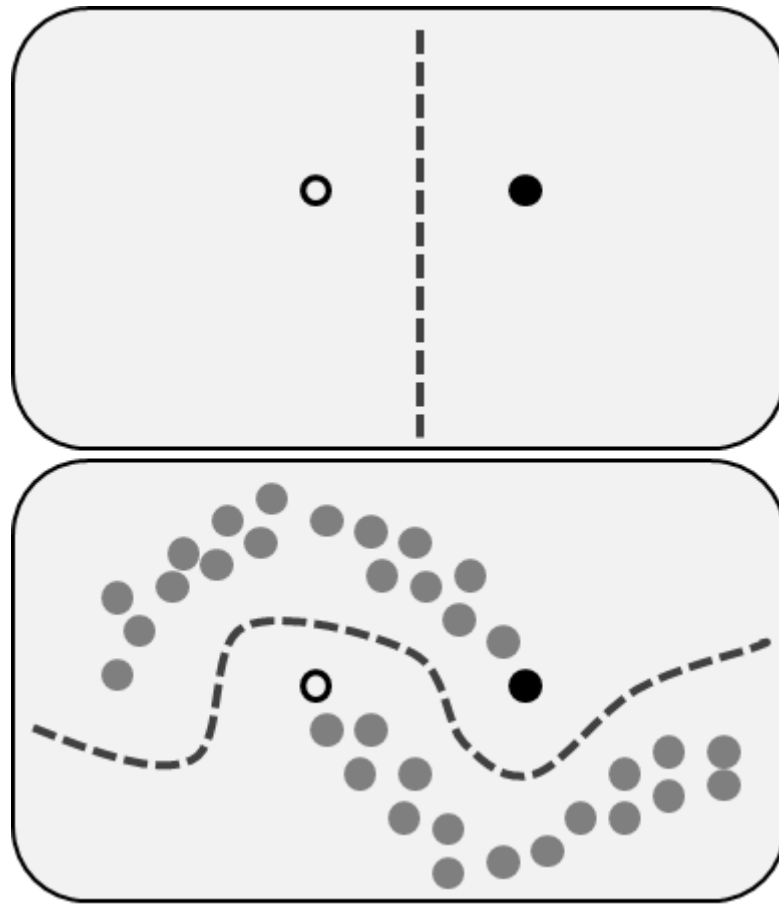
Επιτηρούμενη μάθηση (αλλιώς επιβλεπόμενη μάθηση ή μάθηση με επίβλεψη) (supervised learning): Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο», και ο στόχος είναι να μάθει ένα γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα.

Μη επιτηρούμενη μάθηση (αλλιώς μη επιβλεπόμενη μάθηση ή μάθηση χωρίς επίβλεψη) (unsupervised learning): Χωρίς να παρέχεται κάποια εμπειρία στον αλγόριθμο μάθησης, πρέπει να βρει τη δομή των δεδομένων εισόδου. Η μη επιτηρούμενη μάθηση μπορεί να είναι αυτοσκοπός (ανακαλύπτοντας κρυμμένα μοτίβα σε δεδομένα) ή μέσο για ένα τέλος (χαρακτηριστικό της μάθησης).

Ενισχυτική μάθηση: Ένα πρόγραμμα υπολογιστή αλληλεπιδρά με ένα δυναμικό περιβάλλον στο οποίο πρέπει να επιτευχθεί ένας συγκεκριμένος στόχος (όπως η οδήγηση ενός οχήματος), χωρίς κάποιος δάσκαλος να του λέει ρητά αν έχει φτάσει κοντά στο στόχο του. Ένα άλλο παράδειγμα είναι να μάθει να παίζει ένα παιχνίδι εναντίον κάποιου αντιπάλου.

Μεταξύ της επιτηρούμενης και της μη επιτηρούμενης μάθησης είναι η ημι-επιτηρούμενη μάθηση, όπου ο δάσκαλος δίνει ένα ελλιπές εκπαιδευτικό σήμα: ένα σύνολο εκπαίδευσης με κάποια (συχνά πολλά) από τα αποτελέσματα στόχους να λείπουν. Η μεταγωγή είναι μια ειδική περίπτωση της αρχής αυτής, όπου το σύνολο των καταστάσεων του προβλήματος είναι γνωστό κατά το χρόνο εκμάθησης, όμως ένα μέρος των στόχων λείπουν.

Μία μηχανή διανυσμάτων υποστήριξης, όπου τα δεδομένα ταξινομούνται σε δύο κλάσεις, που χωρίζονται από ένα γραμμικό σύνορο. Εδώ, έχει μάθει να διακρίνει τους μαύρους από τους άσπρους κύκλους. Στο πάνω μέρος της εικόνας 5 δείχνει το αποτέλεσμα αν 'έβλεπε' το μοντέλο μας μονό ένα άσπρο κύκλο και έναν μαύρο κύκλο, και στην κάτω πλευρά βλέπουμε πως αυτό με την εκπαίδευση μπορεί και υιοθετεί σύνθετους τρόπους για τον διαχωρισμό.



Εικόνα 4 – παράδειγμα για διαχωρισμό σημείων (μαύρα και άσπρα)

Μεταξύ άλλων κατηγοριών μηχανικής μάθησης, υπάρχει ακόμα διαδικασία εκμάθησης (meta learning) που μαθαίνει στην μηχανή (να αναπτύσσει) τις δικές της επαγωγικές μεθόδους, βασιζόμενο στην προηγούμενη εμπειρία. Η Αναπτυξιακή μάθηση (Developmental robotics), η οποία έχει ανεπτυχθεί για την εκμάθηση από ρομπότ, δημιουργεί τη δική της ακολουθία μαθησιακών καταστάσεων, ώστε το ρομπότ συσσωρευτικά αποκτά ποικιλία δεξιοτήτων μέσω της αυτόνομης αυτοεξερεύνησης και της κοινωνικής αλληλεπίδρασης με ανθρώπους εκπαιδευτές και χρησιμοποιώντας μηχανισμούς καθοδήγησης, όπως η ενεργητική μάθηση, η ωρίμανση και η μίμηση.

Μια άλλη κατηγοριοποίηση των προβλημάτων μηχανικής μάθησης προκύπτει όταν κάποιος θεωρήσει το επιθυμητό αποτέλεσμα του συστήματος μηχανικής μάθησης.

Στην ταξινόμηση, τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και η μηχανή πρέπει να κατασκευάσει ένα μοντέλο, το οποίο θα αντιστοιχίζει τα δεδομένα σε μία ή περισσότερες κλάσεις, έτσι το πρόβλημα που προκύπτει είναι ο προσδιορισμός της κατηγορίας που ανήκει κάθε νέα παρατήρηση, σε σχέση με ένα αρχικό σετ εκπαίδευσης που έχουμε δημιουργήσει. Τα φίλτρα Spam είναι ένα παράδειγμα ταξινόμησης, όπου οι είσοδοι είναι τα emails ή άλλα μηνύματα και οι κλάσεις είναι "spam" και "όχι spam". Αυτό συνήθως εμπίπτει στην επιτηρούμενη μάθηση.

Στην παλινδρόμηση, χρησιμοποιώντας τεχνικές στατιστικών μοντέλων για την εύρεση συσχετίσεων μεταξύ κάποιας μεταβλητής και κάποιας άλλης ανεξάρτητης από τα δεδομένα

μεταβλητή, χρησιμοποιείται για την εύρεση μεταβλητών πρόβλεψης και προϋποθέτει ότι τα σχετικά δεδομένα που αναλύει ότι ταιριάζουν με μερικά γνωστά είδη συνάρτησης ώστε να παράξει την καλύτερη δυνατή συνάρτηση για την μοντελοποίηση των δεδομένων. Επίσης πρόβλημα επιτηρούμενης μάθησης, τα αποτελέσματα είναι συνεχή και όχι διακριτά.

Στην συσταδοποίηση, ένα σύνολο εισόδων πρόκειται να χωριστεί σε ομάδες. Σε αντίθεση με την ταξινόμηση, οι ομάδες δεν είναι γνωστές εκ των προτέρων, καθιστώντας αυτόν τον διαχωρισμό τυπική εργασία μη επιτηρούμενης μάθησης.

Πέραν όμως της ταξινόμησης τους , ένα πλήρες και αυτόνομο σύστημα διαχείρισης βάσης δεδομένων κατηγοριοποιεί τα δεδομένα και βελτιστοποιεί τους πίνακες δεδομένων του. Ο βασικότερος όμως ρόλος του είναι ότι έχει την υποχρέωση να ελέγχει την ακεραιότητα των εισαγόμενων στοιχείων και την απόδοσή τους με διαφορετικούς τύπους ανάλογα με τις ανάγκες του κάθε χρήστη.

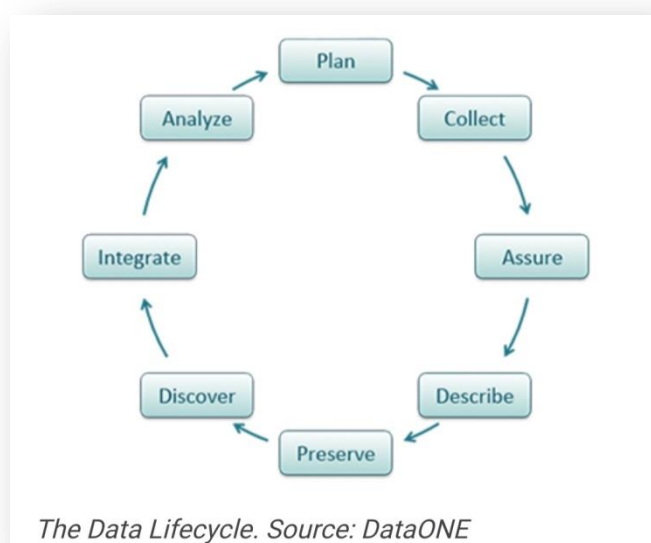


26

4.2 Διαχείριση Big Data

Ο Όρος Big Data αναφέρεται στα δεδομένα που είναι τόσο μεγάλα, γρήγορα ή περίπλοκα τα οποία καθιστούν δύσκολη ή ακόμα και αδύνατη την επεξεργασία τους χρησιμοποιώντας παραδοσιακές τεχνικές διαχειρίσεις δεδομένων. Η διαδικασία πρόσβασης και αποθήκευσης γιγαντίων ποσόν δεδομένων πληροφοριών υπάρχει αρκετά χρονιά, αλλά η χρήση του όρου ξεκίνησε να γίνεται καθημερινή στα τέλη του 2000. [3]

Πιο συγκεκριμένα, η μεγάλη διαχείριση δεδομένων είναι μια γενική έννοια που περιλαμβάνει τις πολιτικές, τις διαδικασίες και την τεχνολογία που χρησιμοποιούνται για τη συλλογή, αποθήκευση, διακυβέρνηση, οργάνωση, διαχείριση και παράδοση μεγάλων όγκων δεδομένων. Ωστόσο ,περιλαμβάνει καθαρισμό δεδομένων, μετανάστευση, ενσωμάτωση και προετοιμασία για χρήση στις αναφορές και στα αναλυτικά στατιστικά στοιχεία. Έχει επίσης την τάση να ταυτίζεται συχνά από το ευρύ κοινό με την εικόνα της διαχείρισης του κύκλου ζωής δεδομένων. Πρόκειται για μια προσέγγιση σχετικά με τον προσδιορισμό των πληροφοριών που πρέπει να αποθηκεύονται στο πλαίσιο της πληροφορικής μιας επιχείρησης καθώς και το γεγονός ότι τα δεδομένα θα μπορούν να διαγράφονται με ασφάλεια όταν και εάν αυτό κριθεί απαραίτητο από τον χρήστη.



Εικόνα 6 – Κύκλος ζωής των δεδομένων

Τα τρία βασικά χαρακτηριστικά των Big Data είναι: [9]

1. Όγκος

Ο όγκος των δεδομένων έχει μεγάλη σημασία, διότι πρέπει να επεξεργαστούμε μεγάλες ποσότητες ασυγκρότητων δεδομένων. Αυτά μπορεί να είναι δεδομένα ,

άγνωστης τιμής όπως είναι η ροή από twitter αναρτήσεις, ροές αιτήσεων σε μια ιστοσελίδα ή αισθητήρες ενεργοποιήσεις.

2. Ταχύτητα

Είναι ο χρόνος στον οποίο τα δεδομένα εισέρχονται στο σύστημα μας η επεξεργάζονται. Κανονικά δεδομένα μεγάλης ταχύτητας εισέρχονται απευθείας στην μνήμη (RAM) και όχι στον δίσκο, έτσι μας επιτρέπεται η επεξεργασία σε πραγματικό χρόνο ή πολύ κοντά σε αυτόν.

3. Διαφορετικότητα

Αναφερόμαστε στους διάφορους τύπους μεταβλητών που είναι στην διάθεση μας. Στα παραδοσιακά συστήματα οι τύποι μεταβλητών είναι κατανομημένα σε συσχετιζόμενες βάσεις δεδομένων, Με το Big Data τα δεδομένα μας έρχονται ως μη δομημένες τιμές. Π.χ. Απλό κείμενο, ήχος, βίντεο, μεταδεδομένα, τιμές αισθητήρα.

4.3 Μηχανική Μάθηση & Δεδομένα

Με την εμφάνιση του IOT , όλο και περισσότερες συσκευές και αντικείμενα είναι στο ιντερνέτ , συλλέγοντας δεδομένα επεξεργάζοντας τα και παράγοντας καινούργια. Αυτό τοποθετήθηκε στον επιστημονικό κόσμο με τον όρο Μηχανική Μάθηση. Η μηχανική μάθηση είναι σκέλος της επέκτασης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Η μηχανική μάθηση απαρτίζεται από τη μελέτη, την ανάπτυξη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν με σκοπό να κατασκευαστούν μοντέλα από πειραματικά στοιχεία και δεδομένα με στόχο να προβλέψουν αποτελέσματα βασισμένα στα δεδομένα ή να λάβουν αποφάσεις που θα εκφράζονται ως το αποτέλεσμα το οποίο αναζητά.



Εικόνα 7 – Μηχανική μάθηση και δεδομένα

Όμως , η μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ακατόρθωτος.

Συμπερασματικά λοιπόν ,με την εξέλιξη της μηχανικής μάθησης είναι πλέον εφικτό να αξιοποιήσουμε όσο καλύτερα γίνεται τις τεράστιες ποσότητες δεδομένων χωρίς να έχουμε κάποιο ρητό πρόγραμμα , έχοντας στην διάθεση μας της τεχνικές προπόνησης που αυτή προσφέρει.

5.1 Αντιπροσωπεύοντας τη γνώση

Λέγοντας γνώση, εννοούμε τις γενικές αλήθειες, οι οποίες αντιπροσωπεύονται από μικρές ομάδες συμβολισμών, όπως είναι τα γράμματα που ο άνθρωπος χρησιμοποιεί, η γλώσσα σώματος ή ακόμα και επιστήμες όπως τα μαθηματικά και η φυσική που έχει δημιουργήσει τρόπους και συναρτήσεις για να εκφράσει την πραγματικότητα χρησιμοποιώντας την λογική. Παραδείγματα της γνώσης είναι :

- **Όλοι οι άνθρωποι είναι θνητοί.**
- **Η ταχύτητα ενός αντικειμένου ισούται με το πηλίκο της απόστασης προς τη μονάδα του χρόνου.**

Από τους αρχαίους έλληνες φιλόσοφους η γνώση έχει εκφραστεί ως κάποιο συλλογισμό, π.χ. κανόνες του τύπου

IF <condition> THEN <conclusion>

εκφράζονται ως:

IF human(x) THEN mortal(x)

Στη μοντέρνα επιστήμη πιο συγκεκριμένα στη φυσική , έχει αποδειχτεί ότι είναι πιο εύκολο να παρουσιαστεί η γνώση μέσα από μια συνάρτηση ή από περιορισμούς ,πιο γενικά από τον τύπο που εκφράζει την σχέση της ταχύτητας (u), του χρόνου (t) και της απόστασης (s):

$$u = s/t$$

ο όποιος μπορεί να χρησιμοποιηθεί για να υπολογίσεις και τα δυο , δηλαδή το u από το s και το t ή το s από το u και το t. Οι περιορισμοί είναι μεμονωμένα πεδία στα οποία μπορούμε να εκφράσουμε τις σχέσεις μεταξύ μεταβλητών τιμών , και να αναθέσουμε ποιοι συνδυασμοί επιτρέπονται και μη.

5.2 Βασικά συστήματα διαχείρισης

Όταν προσπαθούμε να εκφράσουμε τη γνώση με κανόνες, νέα δεδομένα μπορούν να συμπεριληφθούν από ένα λογικό συμπέρασμα που ονομάζεται αφαιρετική μέθοδος. Έχοντας έναν κανόνα από το $A \rightarrow B$ και την παρατήρηση πως το A είναι αληθές, τότε μπορούμε καταλήξουμε στο αποτέλεσμα B . Ο Καθολικός έλεγχος από τα βασικά συστήματα διαχείρισης αλλάζουν είτε προς τα εμπρός είτε προς τα πίσω.

Χρησιμοποιώντας την προς τα εμπρός αλλαγή, ξεκινάμε από τις συνθήκες τις οποίες γνωρίζουμε ότι είναι αληθές, προς τα συμπεράσματα που θέλουμε να εξάγουμε.

Για την προς τα πίσω αλλαγή, κινούμαστε από τα συμπεράσματα που γνωρίζουμε ότι είναι αληθινά (Τους όρους των κανόνων) τα οποία πρέπει να προδίδουν τα συμπεράσματα.

Η διαδικασία της αφαίρεσης είναι λογικά ορθή και ο μονός λόγος για μια συνολικά λανθασμένη ή αντικρουόμενη διαδικασία συλλογιστικής έγκειται στη ανακρίβεια των κανόνων.

Η διασφάλιση της ορθότητας ενός συστήματος που βασίζεται σε κανόνες σε ένα κόσμο που αλλάζει συνεχώς εξακολουθεί να αποτελεί μεγάλο πρόβλημα και ήταν ένα από τα σημαντικότερα εμπόδια στην επιτυχία των συστημάτων εμπειρογνώμων που βασίζεται σε κανόνες. Ο λόγος που γίνεται αυτό είναι ότι οι αφαιρετικοί κανόνες ισοδυναμούν πάντα με μια αναπαράσταση γνώσης που εξαρτάται από το περιβάλλον, καθώς η ορθότητα κάθε κανόνα εξαρτάται από το ποιο άλλοι κανόνες υπάρχουν στην βάση μας.

Για παράδειγμα, εξετάστε το πρόβλημα της διαμόρφωσης ενός αυτοκινήτου χρησιμοποιώντας ένα σύστημα βασισμένο σε κανόνες. Ένα αυτοκίνητο σε αυτό το παράδειγμα μοντελοποιείται χρησιμοποιώντας πέντε μεταβλητές, δηλαδή πακέτο, πλαίσιο, κινητήρα, κιβώτιο ταχυτήτων και τον τύπο. Ο τομέας μιας μεταβλητής είναι το σύνολο των πιθανών τιμών για αυτήν τη μεταβλητή και οι συγκεκριμένες απαιτήσεις των πελατών είναι απλά τιμές για τον τύπο και το πακέτο των μεταβλητών. Οι πέντε μεταβλητές και οι πιθανές τιμές τους περιγράφονται με περισσότερες λεπτομέρειες παρακάτω.

1. Πακέτο με τιμές στάνταρ πολυτέλειας

Η μεταβλητή του πακέτου περιγράφει τον εξοπλισμό του αυτοκινήτου. Το πολυτελές πακέτο θα περιλαμβάνει και ένα κλιματιστικό και ένα κεντρικό σύστημα κλειδώματος. Το πολυτελές πακέτο περιλαμβάνει μόνο το κεντρικό σύστημα κλειδώματος. Κανένα από τα είδη εξοπλισμού δεν περιλαμβάνεται στο τυποποιημένο πακέτο.

2. Ο τύπος με τις τιμές για спор-οικογενειακό αυτοκίνητο.

Το τμήμα μάρκετινγκ αποφάσισε να πουλήσει μόνο σε δύο τμήματα αυτοκινήτων. Καμία παραλαβή ή άλλο είδος αυτοκινήτου δεν μπορεί να πωληθεί εκτός από σπορ αυτοκίνητα και οικογενειακά αυτοκίνητα.

3. Μηχανές με τιμές A & B

Διατίθενται τρεις διαφορετικοί τύποι κινητήρων. Ο κινητήρας A είναι μια σπορ έκδοση, μπορεί επομένως να χρησιμοποιηθεί για το σπορ αυτοκίνητο. Η μηχανή B είναι μια ομαλή έκδοση για λιμουζίνες.

4. Μετάδοση με τιμές χειροκίνητο αυτόματο ημι-αυτόματο.

Οι αυτόματες και ημι-αυτόματες μεταδόσεις βρίσκονται συνήθως σε λιμουζίνες. Οι χειροκίνητες μεταδόσεις χρησιμοποιούνται για αυτοκίνητα εκτός δρόμου και σπορ αυτοκίνητα.

5. Πλαίσιο τιμών μετατρέψιμο σε sedan hatchback

Τα αυτοκίνητα μπορούν να παραχθούν με τρεις διαφορετικούς τύπους πλαισίων. Το πλαίσιο Hatchback και sedan είναι κατάλληλο για οικογένειες. Τα μετατρέψιμα πλαίσια από την άλλη πλευρά χρησιμοποιούνται γενικά για τύπους σπορ.

Η διαδικασία καθορισμού τιμών για τις άλλες μεταβλητές, δεδομένης ορισμένων συγκεκριμένων απαιτήσεων πελατών και της βάσης κανόνων παρακάτω, ονομάζεται διαδικασία διαμόρφωσης. Η διαμόρφωση μπορεί να απεικονιστεί ως ένα είδος λογαριασμού υλικού με πληροφορίες διάταξης.

Όλοι οι κανόνες είναι της μορφής EAN η προϋπόθεση ΤΟ συμπέρασμα. Ο αλγόριθμος «αλυσοειδούς αλυσίδας» επιλέγει τους κανόνες όπου οι χώροι είναι αληθινοί και τους πυροδοτεί σε συμπεράσματα γεγονότων. Γνωρίζοντας για παράδειγμα, ότι ο πελάτης επιθυμεί ένα σπορ αυτοκίνητο ο αλγόριθμος καταλήγει αμέσως από τον κανόνα 8 (βλ. Παρακάτω), ότι το αυτοκίνητο θα έχει χειροκίνητο κιβώτιο ταχυτήτων.

Rule1 IF Package = Deluxe and Frame = convertible THEN Engine A

Rule 2 IF Package = Deluxe and Frame = hatchback THEN Engine B

Rule 3 IF Package = Standard and Frame = convertible THEN Engine A

Rule 4 IF Engine = A THEN Transmission = manual

Rule 5 IF Engine = B THEN Transmission = automatic

Rule6 IF Type = Sportscar THEN Frame = convertible

Rule7 IF Type = Familycar THEN Frame = sedan

Rule8 IF Type = Sportscar THEN Transmission = manual

Λαμβάνοντας υπόψη τις απαιτήσεις του πελάτη "deluxe sportscar" (τιμές για το πακέτο μεταβλητών και τον τύπο) μπορεί κανείς να καθορίσει τη διαμόρφωση με μια διαδικασία μελλοντικής αλυσίδας στους κανόνες 1-8. Η διαμόρφωση ή η έξοδος της διαδικασίας διαμόρφωσης είναι οι πέντε πλειάδες:

Type = sportscar (είσοδος πελάτη)

πακέτο = πολυτελές (είσοδος πελάτη)

καρέ = μετατρέψιμο (βάσει του κανόνα 6 με δεδομένο τον τύπο εισόδου = σπορ αυτοκίνητο)

κινητήρας = A (κατά κανόνα 1 δεδομένο Πακέτο = πολυτελές και πλαίσιο = μετατρέψιμο)

μετάδοση = χειροκίνητο (σύμφωνα με τον κανόνα 8 με δεδομένο τον τύπο εισόδου = σπορ αυτοκίνητο ή με τον κανόνα 4 δώστε τον κινητήρα πραγματικού γεγονότος = A

Η διαδικασία αφαίρεσης σε ένα σύστημα βάσης κανόνων μπορεί να απεικονιστεί χρησιμοποιώντας ένα γράφημα παραγώγων. Ένα βέλος υποδεικνύει την αφαίρεση μιας τιμής για μια μεταβλητή στο τέλος του βέλους που δίνει την προϋπόθεση (μια τιμή για τη μεταβλητή) στην αρχή του βέλους. Είναι δυνατοί οι κύκλοι στις διαδρομές γραφικών ομόσπορων που τελειώνουν στην ίδια κορυφή, επειδή η γνώση μπορεί να προέρχεται από διαφορετικές πηγές. Ο κανόνας 8 στο παραπάνω παράδειγμα μπορεί να προέρχεται από κανονισμούς μάρκετινγκ, ενώ οι κανόνες 4 και 5 καθορίζονται από τεχνικά ζητήματα.

6. Προτεινόμενο Σύστημα

6.1 Περίληψη Συστήματος

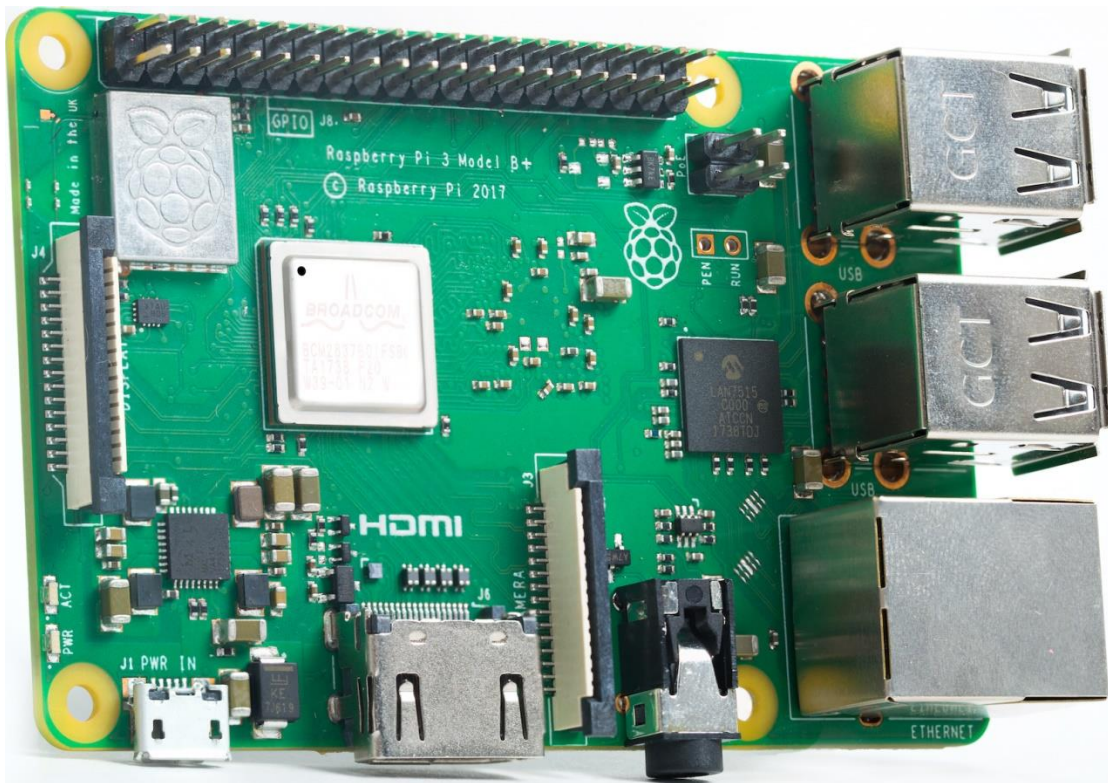
Το προτεινόμενο σύστημα που θα αναλύσουμε είναι ένα Θερμοκήπιο ΙΟΤ. Στόχος της κατασκευής είναι η μοντελοποίηση και μελέτη ενός πραγματικού θερμοκηπίου σε μικρότερη κλίμακα, το οποίο θα αντιδρά σύμφωνα με τις αλλαγές του περιβάλλοντος και θα παίρνει τις ανάλογες αποφάσεις, όπως ακριβώς θα έκανε ένας γεωπόνος στην πραγματικότητα.

Με την χρήση αισθητήρων i)Θερμοκρασίας αέρα και χώματος ii)Υγρασίας του αέρα, το Θερμοκήπιο παράγει ροές δεδομένων (θερμοκρασίας και υγρασίας), τις οποίες το σύστημα επεξεργάζεται χρησιμοποιώντας Μηχανική Μάθηση (Machine learning). Με βάση τις πληροφορίες αυτές είναι δυνατή η πρόβλεψη των μεταβλητών στο κοντινό μέλλον. Σύμφωνα λοιπόν με προκαθορισμένους κανόνες (rule-based system) που θέτει ο γεωπόνος στο σύστημα, λαμβάνονται αποφάσεις ώστε να έχουμε τις καλύτερες δυνατές συνθήκες για την καλλιέργεια των φυτών.

6.2 Υλικό / Hardware

Για την υλοποίηση του project χρησιμοποιήθηκε πλακέτα Raspberry Pi για:

- i) Τη λήψη αποφάσεων
- ii) Τη λήψη δεδομένων από τους αισθητήρες και
- iii) Τη διαχείριση δεδομένων



Εικόνα 8 – Raspberry pi 3 b+ πλακέτα

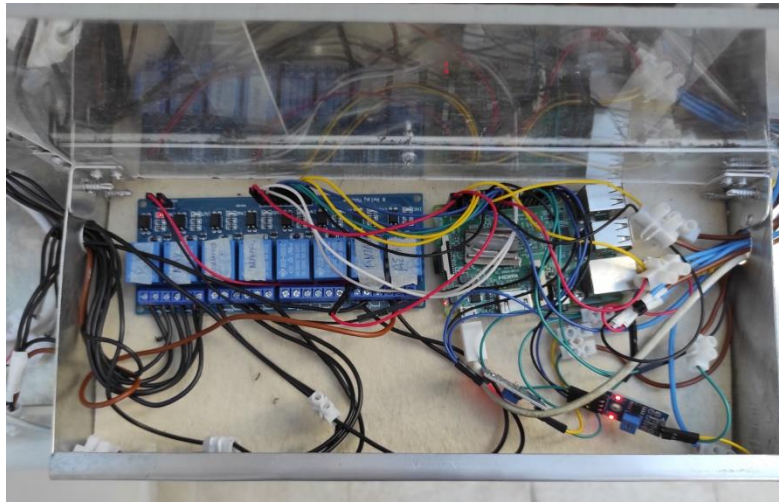
Το Raspberry Pi είναι μια σειρά μικρών υπολογιστών που αναπτύχθηκε στο ΗΒ. Έχει ως σκοπό την εκμάθηση βασικών εννοιών της επιστήμης των υπολογιστών και τη χρήση σε ρομποτική. Χρησιμοποιείται πλέον σε πολλά παιδεία ένα εκ των οποίων είναι το ΙΟΤ. Κριτήρια για την επιλογή του στη δική μας περίπτωση αποτέλεσαν:

- 1)Το χαμηλό του κόστος
- 2)Φορητότητα
- 3)Χαμηλή ενεργειακή κατανάλωση
- 4)Χρησιμοποιεί κώδικα ανοιχτού λογισμικού

Τα τεχνικά του χαρακτηριστικά αναγράφονται παρακάτω.

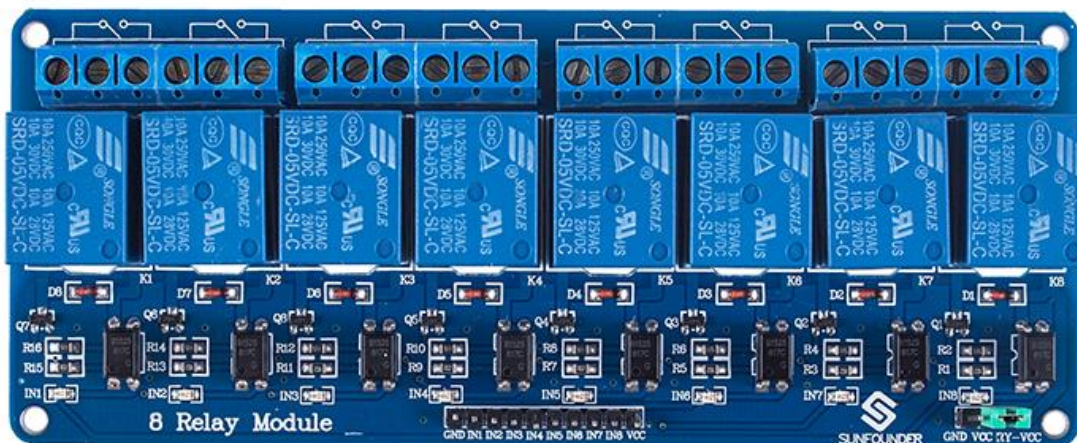
- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

- Dimensions: 82mm x 56mm x 19.5mm, 50g



Εικόνα 9 – raspberry και ρελέ στο θερμικήπιο

Ηλεκτρονόμος (Ρελέ)



Εικόνα 10 – ηλεκτρονόμος 8 relay module

Πρόκειται για έναν ηλεκτρονόμο οκτώ καναλιών με χαμηλή τάση ενεργοποίησης ο οποίος είναι σχεδιασμένος για την καθοδήγηση ηλεκτρικών συσκευών υψηλής ενεργειακής λειτουργίας.

Χαρακτηριστικά Ηλεκτρονόμου:

- Support control of 10A 30V DC and 10A 250V AC signals
- 5V 8-Channel Relay interface board
- LOW level trigger, equipped with indicator easy to recognize the working status
- Each relay has NO and NC ports, easier to connect and control the connected devices
- Selection of plastic material for high temperature and better chemical solution performance.

- PCB size: 5.7 x 13.8 cm

Θύρες Ηλεκτρονόμενου:

- VCC: Connected to positive supply voltage (supply power according to relay voltage)
- GND: Connected to negative supply voltage
- IN1: Signal(LOW Level) triggering terminal 1 of relay module
- IN2: Signal(LOW Level) triggering terminal 2 of relay module
- IN3: Signal(LOW Level) triggering terminal 3 of relay module
- IN4: Signal(LOW Level) triggering terminal 4 of relay module
- IN5: Signal(LOW Level) triggering terminal 5 of relay module
- IN6: Signal(LOW Level) triggering terminal 6 of relay module
- IN7: Signal(LOW Level) triggering terminal 7 of relay module
- IN8: Signal(LOW Level) triggering terminal 8 of relay module

Μέσα από τις αποφάσεις που λαμβάνονται από το πρόγραμμα το Raspberry Pi με κατάλληλα σήματα ελέγχει το ρελέ ώστε τα να ενεργοποιηθούν ή να απενεργοποιηθούν οι λειτουργίες του θερμοκηπίου.

Τα εξαρτήματα που χρησιμοποιούνται για τις λειτουργίες αυτές είναι:

A) Αντλία νερού

B) Ανεμιστήρας

Γ) Μοτέρ για την ανύψωση του καλύμματος

Αναλυτικότερα:

A) Αντλία νερού

Πρόκειται για μια περισταλτική αντλία θετικού εκτοπίσματος που χρησιμοποιείται για την άντληση του νερού. Καθώς το στροφέιο περιστρέφεται, το μέρος του σωλήνα υπό συμπίεση συσφίγγεται, αναγκάζοντας έτσι το υγρό που πρόκειται να αντληθεί, να κινηθεί μέσω του σωλήνα προς την έξοδο.

Χαρακτηριστικά:

- Voltage: 12V
- Current: 1A
- Power: 5W
- Noise level: 40 dB
- Flow rate range: 5.2ml/min - 90ml/min
- Flow rate:
 - 1x3 Silicone tube: >11ml/min
 - 2x4 Silicone tube: >39ml/min

3x5 Silicone tube: >80ml/min

- Rotate speed: 1-500 rpm
- Pulse: Three rollers, Low pulse
- Pump head: Engineering plastic
- Pump Pipe: Transparent silicone pipe
- Working environment: Temperature 0-40°C
- Relative humidity: <80%
- Color: Black/Green
- Driver size: $\varnothing 27.6 \times 37.9 \text{ mm}$
- Powerhead size: $\varnothing 31.7 \times 20 \text{ mm}$
- Silica gel tube: 2x4mm (IDxOD)
- Weight: 110g



Εικόνα 11 – Αντλία νερού και ντεπόζιτο αποθήκευσης νερού

B) Ανεμιστήρας:

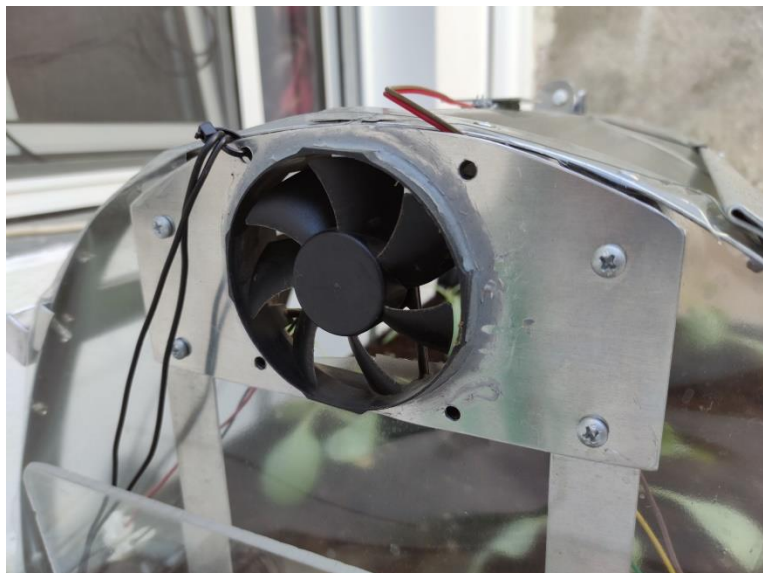
Πρόκειται για ένα περιστρεφόμενο πτερυγωτό μηχανήμα που επιταχύνει τον εξωτερικό αέρα και τον εισάγει στο θερμοκήπιο. Ως αποτέλεσμα έχουμε την αφύγρανση και την μείωση της θερμοκρασίας του αέρα στο περιβάλλον του μοντέλου μας.

Χαρακτηριστικά:

- Voltage: 12V
- Radius: 80mm
- Nominal Speed: 1200rpm



Εικόνα 12 – Ανεμιστήρας 80mm



Εικόνα 13 – Ανεμιστήρας στο θερμοκήπιο

Γ) Μοτέρ

Πρόκειται για μια ηλεκτροκίνητη συσκευή που περιστρέφει έναν άξονα στον οποίο είναι κολλημένη η μια άκρη ενός σχοινιού συνεπώς το σχοινί τυλίγεται. Η άλλη άκρη του οποίου είναι δεμένη στο κάλυμμα του θερμοκηπίου το οποίο ανοίγουμε όταν θέλουμε να εμπλουτίσουμε το περιβάλλον μας με φρέσκο αέρα καθώς και για τη ψύξη του χώματος.



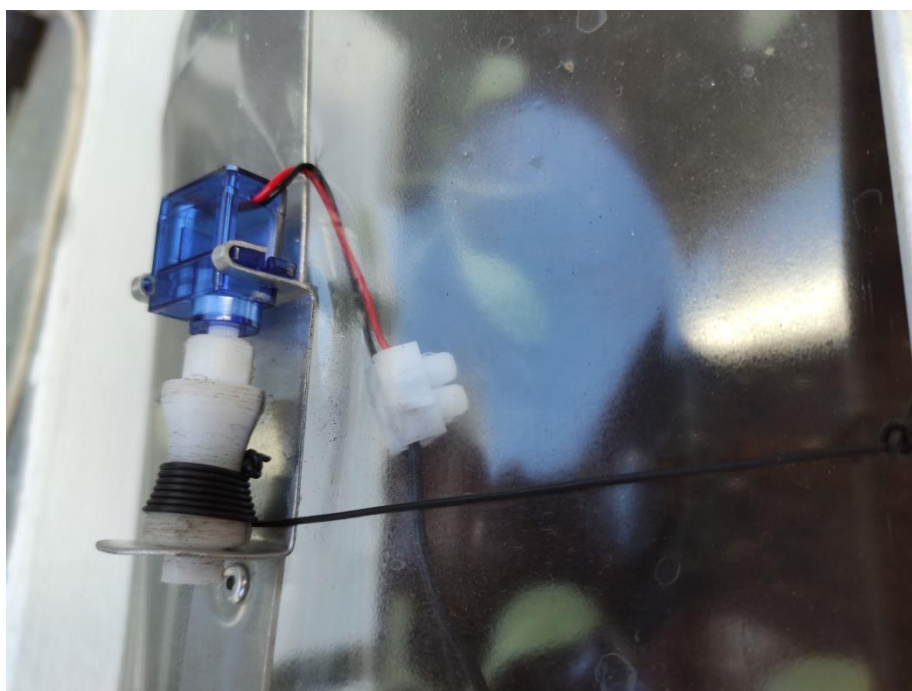
Εικόνα 14 – Μοτέρ

Χαρακτηριστικά:

- Spline Count: 21
- No load speed:
- 110RPM (4.8v)
- 130RPM (6v)
- Running Current (at no load):
- 100mA (4.8v)
- 120mA (6v)
- Peak Stall Torque (4.8v): 1.3 kg.cm / 18.09 oz.in
- Peak Stall Torque (6v): 1.5 kg.cm / 20.86 oz.in
- Stall Current:
- 550mA (4.8v)
- 650mA (6v)
- Διαστάσεις: 32.3mm x 12.3mm x 29.9mm / 1.3" x 0.49" x 1.2"
- Βάρος: 9g



Εικόνα 15 – Χρήση του μοτέρ για ανύψωση της πλαγιάς τέντας



Εικόνα 16 – Μοτέρ στο θερμοκήπιο

Αισθητήρες

Για τη δημιουργία των ροών δεδομένων χρησιμοποιήθηκε ο αισθητήρας DHT11 ο οποίος είναι ένας βασικός χαμηλής κατανάλωσης ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας.

Χρησιμοποιεί αισθητήρα χωρητικότητας για την υγρασία και ένα θερμίστορ που θερμομετρεί τον αέρα του περιβάλλοντος. Στη συνέχεια μετατρέπει αυτές τις πληροφορίες σε ψηφιακό σήμα και τις τροφοδοτεί στο Raspberry Pi.

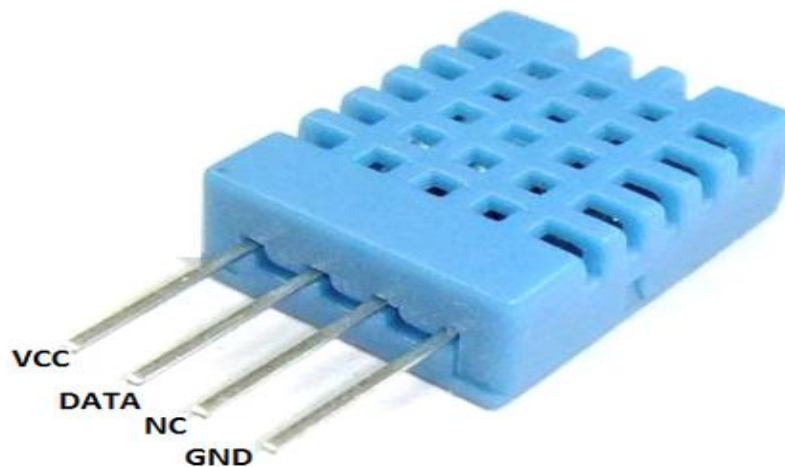
Χαρακτηριστικά:

- Breadboard Friendly : Yes
- Sensor Type:
- Humidity
- Temperature
- Typical Input Voltage: 3VDC

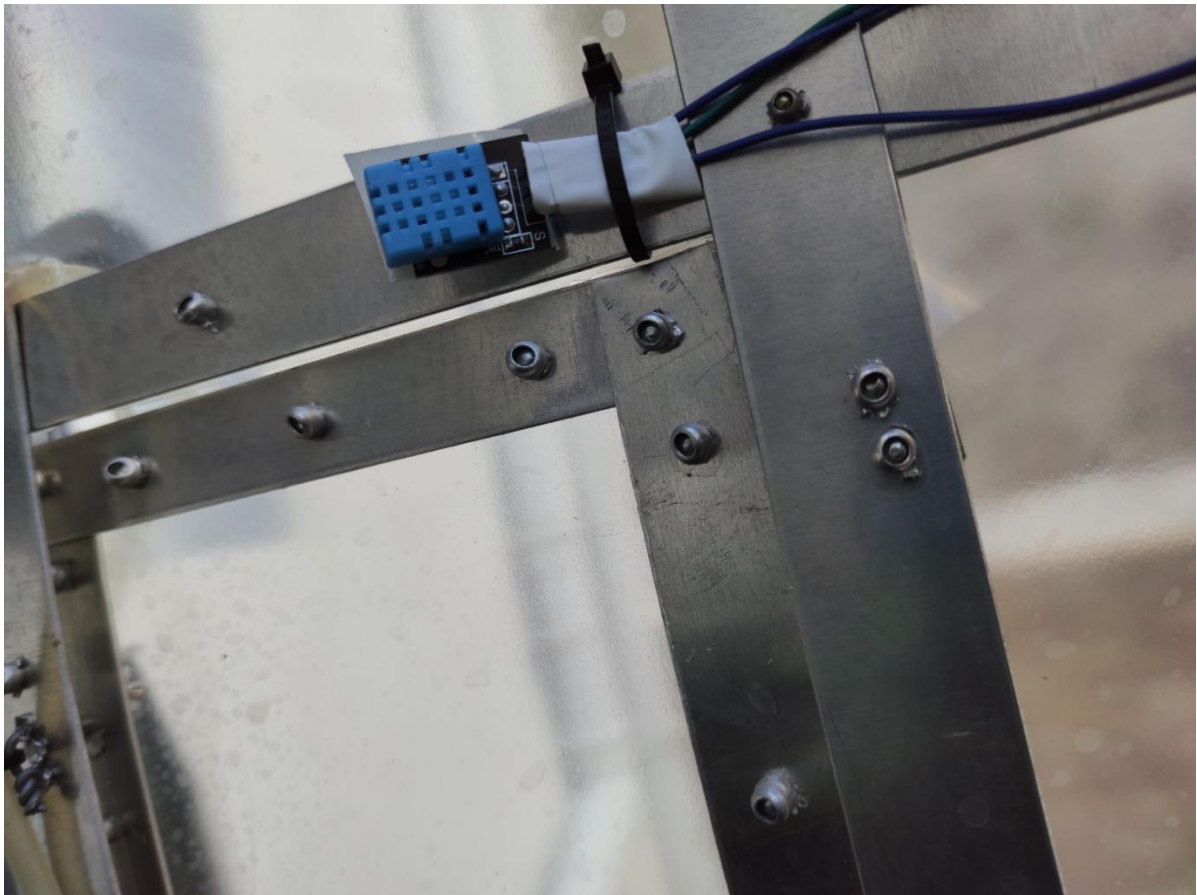
3.3VDC

5VDC

- Operating Current: 2.5mA
- Interface: Digital
- Communication Protocol: Without
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings $\pm 2^{\circ}\text{C}$ accuracy
- No more than 1 Hz sampling rate (once every second)
- Διαστάσεις: 15.5 x 12 x 5.5mm
- Pitch of pins: 0.1"/2.54mm



Εικόνα 17 – Αισθητήρας θερμοκρασίας και υγρασίας



Εικόνα 18 – Αισθητήρας θερμοκρασίας και υγρασίας στο θερμοκήπιο

6.3 Λογισμικό / Software

6.3.1 Αισθητήρες

Για να γίνει εφικτή η επικοινωνία των αισθητήρων με το Raspberry , χρησιμοποιήθηκε η βιβλιοθήκη Adafruit_DHT , η οποία είναι η προτεινομένη βιβλιοθήκη για το συγκεκριμένο αισθητήρα και είναι γραμμένη από τον κατασκευαστή. Μέσω συναρτήσεων της βιβλιοθήκης μπορούμε να διαβάσουμε την τιμή του αισθητήρα.

1. `#!/usr/bin/python3`
2. `import datetime`
3. `import sys`
4. `import Adafruit_DHT`

```
5. import RPi.GPIO as GPIO
6. import time
```

```
1. #Temp sensor
2. def Tempmeasure():
3.     humidity, temperature = Adafruit_DHT.read_retry(11, 4)
4.     #print('Temp: {0:0.1f} C Humidity: {1:0.1f} %'.format(temperature, humidity))
5.     return temperature, humidity
6. #Temp END
7.
8.
9. #Soil Temp sensor
10. def SoilTempmeasure():
11.     soilhumidity, soiltemperature = Adafruit_DHT.read_retry(11, 3)
12.     #print('Soil Temp: {0:0.1f} C Soil Humidity: {1:0.1f} %'.format(soiltemperature, soilhumidity))
13.     return soiltemperature, soilhumidity
14. #Soil Temp END
```

Έχοντας στην διάθεση τις συναρτήσεις με τις οποίες μπορούμε να διαβάσουμε Θερμοκρασία αέρα, Θερμοκρασία χώματος και υγρασία αέρα. Με τη χρήση των βιβλιοθηκών `time` και `datetime`, δημιουργούμε συνάρτηση η οποία θα παίρνει τις τιμές των αισθητήρων ανά λεπτό και θα τις αποθηκεύει σε αρχείο με όνομα την ημέρα που έγινε η μέτρηση ελέγχοντας για λάθος τιμές. Στο τέλος της κάθε μέρας θα γίνεται επαναφορά και η συνάρτηση θα τρέχει από την αρχή, με όνομα εκείνο που αντιστοιχεί στη νέα μέρα.

```
1. if __name__ == '__main__':
2.     try:
3.         while 1:
4.             t = datetime.datetime.today()
5.             ymd = "Data/" + str(datetime.date.today()) + ".txt"
6.             datastream = open(ymd, "a")
7.             #datastream.write(str(datetime.datetime.now()) + "\n")
8.             datastream.close()
9.             count = 0
10.            while 1:
11.                #main programm
12.                datastream = open(ymd, "a")
13.                temperature, humidity = Tempmeasure()
```

```

14.         soiltemperature, soilhumidity = SoilTempmeasure()
15.         data = str(temperature) + "," + str(humidity) + "," + str(soiltemperatur
           e) + "\n"
16.         if not ((temperature is None) | (humidity is None) | (soiltemperature i
           s None)):
17.             if not (((temperature <= -
               1) | (humidity >= 100) | (soiltemperature <= -
               1) | (str(temperature) == 'None') | (str(soiltemperature) == "None"))):
18.                 datastream.write(data)
19.                 datastream.close()
20.                 print("THREAD WRITING FILE :: " + str(datetime.date.today(
                   )) + " :: Temp:" + str(temperature) + "C : Humidity:" + str(humidity) + "% : Soi
                   l Temperature:" + str(soiltemperature) + "C")
21.                 time.sleep(60)
22.                 if ((datetime.datetime.now() > datetime.datetime(t.year,t.month,t.day,
                   23,54,
                   0)) and (datetime.datetime.now() < datetime.datetime(t.year,t.month,t.day,23,5
                   9,59))):
23.                     time.sleep(380)
24.                     break
25.         GPIO.cleanup()
26.     except KeyboardInterrupt:
27.         datastream.close()
28.         GPIO.cleanup()

```

Αυτό το κομμάτι κώδικα αποθηκεύεται σαν αρχείο Sensors.py , το οποίο μπορούμε μετά να καλέσουμε από το βασικό πρόγραμμα ως νήμα.

```

THREAD WRITING FILE :: 2020-10-01 :: Temp:24.0C : Humidity:78.0% : Soil Temperature:23.0C
THREAD WRITING FILE :: 2020-10-01 :: Temp:24.0C : Humidity:78.0% : Soil Temperature:22.0C
THREAD WRITING FILE :: 2020-10-01 :: Temp:24.0C : Humidity:78.0% : Soil Temperature:23.0C
THREAD WRITING FILE :: 2020-10-01 :: Temp:24.0C : Humidity:78.0% : Soil Temperature:23.0C

```

6.3.2 Machine Learning LSTM

Στην συνέχεια το βασικό πρόγραμμα, δημιουργεί πινάκες για τις μεταβλητές μας που θα γεμίσει διαβάζοντας το Dataset , το οποίο αποτελείται από πραγματικές τιμές προηγούμενων περιόδων του θερμοκηπίου ώστε να προπονηθεί το Machine Learning.

```

1. print("MAIN PROGRAMM :: STARTING SETUP")
2. ATemperature = []
3. ASoilTemperature = []
4. AHumidity = []
5. print("MAIN PROGRAMM :: Reading File RDataset for training")
6. with open("Data/RDataset.txt") as file:
7.     for line in file:
8.         line = line.split(",")
9.         ATemperature.append(float(line[0]))
10.        ASoilTemperature.append(float(line[1]))
11.        AHumidity.append(float(line[2]))

```

Χρησιμοποιώντας την βιβλιοθήκη NumPy, ξεχωριστά για κάθε μεταβλητή του Datastream μας, εξάγουμε την τυπική απόκλιση του δείγματος μας καθώς και τον μέσο όρο του.

Έτσι δημιουργούμε μια κανονική κατανομή.

```

1. Temperature_Deviation = np.std(ATemperature)
2. Temperature_Mean = np.mean(ATemperature)
3. Values = 500
4. # print(data)
5. # print(Temperature_Deviation)
6. # print(Temperature_Mean)
7. raw_seq1 = np.random.normal(
8.     Temperature_Mean, Temperature_Deviation, Values)
9. # print(raw_seq1)

```

Βιβλιοθήκες:

```

1. from numpy import array
2. from keras.models import Sequential
3. from keras.layers import LSTM
4. from keras.layers import Dense
5. import tensorflow as tf
6. tf.get_logger().setLevel('FATAL')

```


Χρησιμοποιώντας τη βιβλιοθήκη Keras, η οποία περιέχει τις βιβλιοθήκες Μηχανικής μάθησης ανοιχτού κώδικα του Tensorflow και έχει σχεδιαστεί για την υλοποίηση και την εκπαίδευση των μοντέλων μας.

Δημιουργούμε ένα σύστημα LSTM (Long Short-term Memory), δηλαδή ένα επαναλαμβανόμενο νευρωνικό δίκτυο με το οποίο μπορούμε να κατηγοριοποιήσουμε και να επεξεργαστούμε τα δεδομένα μας με βάση το χρόνο. Αυτό θα γίνει με τη χρήση optimizer τύπου Adam για να επιτύχουμε στοχαστική βελτιστοποίηση παράγοντας 500 τυχαίες τιμές αλλά και χάνσιμο τύπου MSE (Mean squared error), εφόσον βέβαια γνωρίζουμε ότι τα δεδομένα μας δεν έχουν λάθη.

```
1. max1 = -1.0
2. min1 = 99999.0
3. for i in range(Values):
4.     if raw_seq1[i] > max1:
5.         max1 = raw_seq1[i]
6.     if raw_seq1[i] < min1:
7.         min1 = raw_seq1[i]
8.
9. for i in range(Values):
10.    raw_seq1[i] = (raw_seq1[i] - min1) / (max1 - min1)
11.
12.# print(raw_seq1)
13.n_steps = 3
14.# split into samples
15.X, y = split_sequence(raw_seq1, n_steps)
16.# reshape from [samples, timesteps] into [samples, timesteps, features]
17.n_features = 1
18.X = X.reshape((X.shape[0], X.shape[1], n_features))
19.# define model
20.model1 = Sequential()
21.model1.add(LSTM(50, activation='relu', input_shape=(n_steps, n_features)))
22.model1.add(Dense(1))
23.model1.compile(optimizer='adam', loss='mse')
24.# fit model
25.start = datetime.datetime.now()
26.model1.fit(X, y, epochs=10, verbose=0)
27.end = datetime.datetime.now()
28.totalTime = end - start
29.print("MAIN PROGRAMM :: Training time : ", totalTime, " seconds")
```

Την ίδια διαδικασία ακολουθούμε και για τις άλλες δυο ροές δεδομένων.

Τέλος, το βασικό πρόγραμμα παίρνει ανά τρία λεπτά και με αύξουσα χρονική σειρά τις τρεις τελευταίες τιμές της μεταβλητής μας.

```
1. def tail(file, n=3, bs=1024):
2.     f = open(file)
3.     f.seek(0, 2)
4.     l = 1-f.read(1).count("\n")
5.     B = f.tell()
6.     while n >= l and B > 0:
7.         block = min(bs, B)
8.         B -= block
9.         f.seek(B, 0)
10.        l += f.read(block).count("\n")
11.    f.seek(B, 0)
12.    l = min(l, n)
13.    lines = f.readlines()[-l:]
14.    f.close()
15.    return lines
```

Σε αυτό το σημείο γίνεται πρόβλεψη της τιμής στο μέλλον.

```
1. Temperature_last3 = [float(sample1[0]), float(sample2[0]), float(sample3[0])]
2. Temperature_last3_Mean = np.mean(Temperature_last3)
3. Temp_input = array([Temperature_last3[0]-
    Temperature_last3_Mean, Temperature_last3[1]-
    Temperature_last3_Mean, Temperature_last3[2]-Temperature_last3_Mean])
4. Temp_input = Temp_input.reshape((1, n_steps, n_features))
5. Temp_Predict = model1.predict(Temp_input, verbose=0)
6. T1 = float(Temp_Predict[0] + Temperature_last3_Mean)
7. print("MAIN PROGRAMM :: Predict for Temperature : ",T1)
8.
9. SoilTemperature_last3 = [float(sample1[2]), float(sample2[2]), float(sample3[2
    ])]
10. SoilTemperature_last3_Mean = np.mean(SoilTemperature_last3)
11. SoilTemp_input = array([SoilTemperature_last3[0]-
    SoilTemperature_last3_Mean, SoilTemperature_last3[1]-
    SoilTemperature_last3_Mean, SoilTemperature_last3[2]-
    SoilTemperature_last3_Mean])
12. SoilTemp_input = SoilTemp_input.reshape((1, n_steps, n_features))
13. SoilTemp_Predict = model2.predict(SoilTemp_input, verbose=0)
14. S2 = float(SoilTemp_Predict[0] + SoilTemperature_last3_Mean)
15. print("MAIN PROGRAMM :: Predict for Soil Temperature : ",S2)
```



```

16.
17.Humidity_last3 = [float(sample1[1]), float(sample2[1]), float(sample3[1])]
18.Humidity_last3_Mean = np.mean(Humidity_last3)
19.Humidity_input = array([Humidity_last3[0]-
    Humidity_last3_Mean, Humidity_last3[1]-
    Humidity_last3_Mean, Humidity_last3[2]-Humidity_last3_Mean])
20.Humidity_input = Humidity_input.reshape((1, n_steps, n_features))
21.Humidity_Predict = model3.predict(Humidity_input,verbose=0)
22.H3 = float(Humidity_Predict[0] + Humidity_last3_Mean)
23.print("MAIN PROGRAMM :: Predict for Humidity : ",H3)

```

```

MAIN PROGRAMM :: Predict for Temperature : 26.313983917236328
MAIN PROGRAMM :: Predict for Soil Temperature : 24.25456428527832
MAIN PROGRAMM :: Predict for Humidity : 80.62420654296875

```

6.3.3 Rule Based System(Durable)

Βασισμένοι στις προβλέψεις που έχει παράξει το LSTM σύστημα μας και έχοντας από πριν κάποιους κανόνες σε μια βάση δεδομένων , μπορούμε με την κατασκευή ενός Rule Based System να δώσουμε κάποιους κανόνες όπως ακριβώς θα όριζε ένας γεωπόνος . Στην περίπτωση μας θα ορίσουμε τα λαχανάκια Βρυξελών.

Ποιο αναλυτικά θέτουμε κάποιες στοιχειά τα οποία ισχύουν πάντα για τα συγκεκριμένα φυτά.

	Θερμοκρασία Εδάφους	Θερμοκρασία Αέρα	Υγρασία
Ιδανικό	10 -24	17 - 30	30 -65
Καλό	24 -30	23 -30	-
ΟΧΙ κάτω	10	17	0 - 30
ΟΧΙ άνω	30	30	65

Για τη διαδικασία αυτή το σύστημα αποφάσεων που θα χρησιμοποιηθεί είναι το Durable!

```

1. Res = []
2. with ruleset('cabbage'):

```

```

3.     @when_all(m.SoilTemp >= 30.0)
4.     def FirstRule(c):
5.         #print ('Rule1 is Activated : with {0}'.format(c.m.SoilTemp
6.         ))
7.         Res.append(1)
8.     @when_all((m.Temp <= 17.0) | ((m.Humidity >= 0) & (m.Humidity <
9.         = 30.0)))
10.    def SecondRule(c):
11.        #print ('Rule2 is Activated : with {0} {1} {2}'.format(c.m.
12.        SoilTemp,c.m.Temp,c.m.Humidity))
13.        Res.append(2)
14.    @when_all(((m.SoilTemp >= 10.0) & (m.SoilTemp <= 24.0)) | (((m.
15.        Temp >= 17.0) & (m.Temp <= 23.0))) | ((m.Humidity >= 30.0) & (Humid
16.        ity <= 65.0)))
17.    def ThirdRule(c):
18.        #print ('Rule3 is Activated : with {0} {1} {2}'.format(c.m.
19.        SoilTemp,c.m.Temp,c.m.Humidity))
20.        Res.append(3)
21.    @when_all(((m.Temp >= 23.0) & (m.Temp <= 30.0)) | ((m.SoilTemp
22.        >= 24.0) & (m.SoilTemp <= 30.0)))
23.    def FourthRule(c):
24.        #print ('Rule4 is Activated : with {0} {1}'.format(c.m.Soil
25.        Temp,c.m.Temp))
26.        Res.append(4)
27.    @when_all(((m.Humidity >= 65.0) & (m.Humidity <= 100)) | (m.Tem
28.        p >= 30.0))
29.    def FifthRule(c):
30.        #print ('Rule5 is Activated : with {0} {1}'.format(c.m.Humi
31.        dity, c.m.Temp))
32.        Res.append(5)
33.
34.
35. def First():
36.     #print("First")
37.     Pump_on(IN2)
38.     time.sleep(10)
39.     Pump_off(IN2)
40.
41. def Second():
42.     #print("Second")
43.     Reverser_off(IN7,IN8)
44.     Air_off(IN1)
45.
46. def Third():
47.     #print("Third")
48.     Air_off(IN1)

```

```

44.
45.
46. def Fourth():
47.     #print("Fourth")
48.     Reverser_on(IN7,IN8)
49.
50.
51. def Fifth():
52.     #print("Fifth")
53.     Air_on(IN1)
54.     Reverser_on(IN7,IN8)
55.
56. def Result(S, T, H):
57.     global Res
58.     post('cabbage', {'SoilTemp' : S})
59.     post('cabbage', {'Temp' : T})
60.     post('cabbage', {'Humidity' : H})
61.     res = []
62.     for i in Res:
63.         if i not in res:
64.             res.append(i)
65.     for i in res:
66.         if i == 1:
67.             First()
68.         elif i == 2:
69.             Second()
70.         elif i == 3:
71.             Third()
72.         elif i == 4:
73.             Fourth()
74.         elif i == 5:
75.             Fifth()
76.     #print(Res)
77.     #print(res)
78.     Res = []

```

Έτσι το σύστημα επιδιώκει πάντα να πραγματοποιήσει τις “τέλειες” συνθήκες περιβάλλοντος του θερμοκηπίου για το υποκείμενο φυτό.

6.3.4 Υλικές Λειτουργίες

Κάθε φορά που το σύστημα μας παίρνει μια απόφαση , ενεργοποιούνται και οι ανάλογες λειτουργίες, οι οποίες υλοποιούνται μέσω κατάλληλων σημάτων που στέλνει το Raspberry pi στο ρελέ.

A) Ανεμιστήρας:

```
1. def Air_off(pin1):
2.     global Air_Global
3.     if (Air_Global == 1):
4.         GPIO.output(pin1, GPIO.HIGH)
5.         print("FAN :: DEACTIVATED")
6.         Air_Global = 0
7.
8. def Air_on(pin1):
9.     global Air_Global
10.    if (Air_Global == 0):
11.        GPIO.output(pin1, GPIO.LOW)
12.        print("FAN :: ACTIVATED")
13.        Air_Global = 1
14. #Fan END
```

B) Μοτέρ

```
1. def Reverser_on(pin1, pin2):
2.     global Reverser_Global
3.     if (Reverser_Global == 0):
4.         GPIO.output(pin1, GPIO.HIGH)
5.         time.sleep(0.00001)
6.         GPIO.output(pin2, GPIO.HIGH)
7.         print("RELAY REVERSER :: ACTIVATED")
8.         time.sleep(3.4)
9.         breakrotation(pin1, pin2)
10.        Reverser_Global = 1
11.
12. def Reverser_off(pin1, pin2):
13.     global Reverser_Global
14.     if (Reverser_Global == 1):
15.         GPIO.output(pin1, GPIO.LOW)
16.         time.sleep(0.00001)
17.         GPIO.output(pin2, GPIO.LOW)
18.         print("RELAY REVERSER :: ACTIVATED")
19.         time.sleep(3.3)
20.         breakrotation(pin1, pin2)
21.         Reverser_Global = 0
22.
23. def breakrotation(pin1, pin2):
24.     GPIO.output(pin1, GPIO.HIGH)
25.     time.sleep(0.00001)
26.     GPIO.output(pin2, GPIO.LOW)
27.     print("RELAY REVERSER :: STOP")
28. #reverse MOTOR END
```

Γ) Αντλία νερού

```
1. def Pump_off(pin1):
```

```

2. GPIO.output(pin1, GPIO.HIGH)
3. print("PUMP :: DEACTIVATED")
4.
5. def Pump_on(pin1):
6.     GPIO.output(pin1, GPIO.LOW)
7.     print("PUMP :: ACTIVATED")
8. #Pump MOTOR END

```

6.3.5 Αυτόματο Πότισμα

Σε κάθε εποχή οι ανάγκες της χλωρίδας για νερό διαφέρουν, έτσι για τον προσδιορισμό του κατάλληλου ποτίσματος συντάξαμε έναν αλγόριθμο, καθήκον του οποίου αποτελεί ο προσδιορισμός της εποχής που έχουμε. Για την επίτευξη αυτού ο αλγόριθμος διαβάζει την ημερομηνία και ξεχωρίζει το μήνα και άρα την εποχή. Οι ανάγκες του φυτού κατηγοριοποιούνται σε 3 διαφορετικές απαιτήσεις σε νερό, τις χειμερινές, τις καλοκαιρινές και τις ανοιξιάτικες. Υπό τις παραδοχές:

i) Ο Οκτώβριος και ο Νοέμβριος είναι χειμερινοί μήνες.

ii) Ο Σεπτέμβριος είναι καλοκαιρινός.

Σύμφωνα πάντα με τις απαιτήσεις τους ώστε να επιτευχθούν οι 'τέλειες συνθήκες'. Ειδικότερα, το καλοκαίρι απαιτείται μεγαλύτερος όγκος νερού σε πιο πυκνά χρονικά διαστήματα ως αποτέλεσμα της υψηλής θερμοκρασίας και των μη συχνών βροχοπτώσεων που συνεπάγεται και την χαμηλότερη υγρασία στο χώρο. Αντίθετα, το χειμώνα οι απαιτήσεις σε νερό φθίνουν και έτσι ποτίζουμε πιο αραιά και με λιγότερη ποσότητα. Τέλος, η άνοιξη αποτελεί σημείο τομής των δυο προηγούμενων εποχών, δηλαδή οι απαιτήσεις διασταυρώνονται (μεγάλη ποσότητα ανά μεγάλα διαστήματα και μικρή ποσότητα συχνά).

```

1. if __name__ == '__main__':
2.     α:
3.         while 1:
4.             if (datetime.datetime.now().month in [1, 2, 10, 11, 12]):
5.                 t = datetime.datetime.today()
6.                 if (datetime.datetime.now() < datetime.datetime(t.year,t.month,t.day,12,0,0)):
7.                     print("THREAD PUMP :: WINTER :: Module Loaded")
8.                     pause.until(datetime.datetime(t.year,t.month,t.day,12,0))
9.                     Winter()
10.            else:
11.                pause.until(datetime.datetime(t.year,t.month,t.day+1,1,0,0))

```

```

12.     if (datetime.datetime.now().month in [3, 4, 5]):
13.         t = datetime.datetime.today()
14.         if (datetime.datetime.now() < datetime.datetime(t.year,t.month,t.day,12,0,0)):
15.             print("THREAD PUMP :: SPRING :: Module Loaded")
16.             pause.until(datetime.datetime(t.year,t.month,t.day,10,0))
17.             Spring()
18.         else:
19.             pause.until(datetime.datetime(t.year,t.month,t.day+1,1,0,0))
20.         if (datetime.datetime.now().month in [6, 7, 8, 9]):
21.             t = datetime.datetime.today()
22.             if (datetime.datetime.now() < datetime.datetime(t.year,t.month,t.day,7,0,0)):
23.                 print("THREAD PUMP :: SUMMER :: Module Loaded")
24.                 pause.until(datetime.datetime(t.year,t.month,t.day,7,0,0))
25.                 Summer()
26.             else:
27.                 pause.until(datetime.datetime(t.year,t.month,t.day+1,1,0,0))
28.     except KeyboardInterrupt:
29.         GPIO.cleanup()

```

Μετά την αναγνώριση της εποχής επιλέγεται η κατάλληλη συνάρτηση που περιγράφει τον τρόπο ποτίσματος. Μεταβλητές της συνάρτησης αποτελούν οι χρόνοι μεταξύ δύο διαδοχικών ποτισμάτων αν υπάρχει και η ποσότητα του νερού [=]ml.

```

1. def Winter():
2.     ml = 10000
3.     for i in range(0,4):
4.         print("THREAD PUMP :: Opening for " + str(ml/5) + " ml")
5.         Pump_on(IN2)
6.         time.sleep(0.08*ml/10)
7.         Pump_off(IN2)
8.         time.sleep(3600)
9.
10. def Spring():
11.     ml = 10000
12.     for i in range(0,4):
13.         print("THREAD PUMP :: Opening for " + str(ml/5) + " ml")
14.         Pump_on(IN2)
15.         time.sleep(0.08*ml/12)
16.         Pump_off(IN2)
17.         time.sleep(3600)
18.         time.sleep(28800)
19.         print("THREAD PUMP :: Opening for " + str(ml/10) + " ml")
20.         Pump_on(IN2)
21.         time.sleep(0.08*ml/20)
22.         Pump_off(IN2)
23.
24. def Summer():
25.     ml = 12000
26.     for i in range(0,4):

```

```
27.     print("THREAD PUMP :: Opening for " + str(ml/10) + " ml")
28.     Pump_on(IN2)
29.     time.sleep(0.08*ml/20)
30.     Pump_off(IN2)
31.     time.sleep(3600)
32.
33.     time.sleep(25200)
34.     for i in range(0,4):
35.         print("THREAD PUMP :: Opening for " + str(ml/10) + " ml")
36.         Pump_on(IN2)
37.         time.sleep(0.08*ml/20)
38.         Pump_off(IN2)
39.         time.sleep(3600)
```

Έτσι εξασφαλίζουμε ότι τα φυτά παίρνουν πάντα τη σωστή ποσότητα νερού.

Το υποπρόγραμμα καλείται από το βασικό πρόγραμμα ως νήμα και τα δύο τρέχουν παράλληλα.

7.1 Συμπεράσματα

Το Προτεινόμενο Σύστημα που παρουσιάστηκε παραπάνω αποτελεί ένα παραγωγικό μέσω με σκοπό την μαζική παραγωγή και καλλιέργεια φυτών, καθώς έχει την δυνατότητα να συντηρήσει αυτόνομα και αξιόπιστα την συγκομιδή του θερμοκηπίου. Στο σύστημα αυτό αναπτύχθηκε ένα νευρωνικό δίκτυο των οποίων οι τρεις νευρώνες του είναι δομημένοι σε επίπεδα (νήματα) και η εκπαίδευσή (Machine Learning) τους γίνεται με εποπτεία και σε συνδυασμό με τα ερεθίσματα που δέχονται από τους αισθητήρες για την πρόβλεψη των περιβαλλοντικών συνθηκών, ανέπτυξε την ικανότητα να λαμβάνει μέτρα για την φροντίδα των φυτών καθ'όλη την διάρκεια του έτους. Αυτό έχει ως αποτέλεσμα να καθιστά την εργασία του γεωπόνου ευκολότερη και συνεπώς αποδοτικότερη καθώς δεν απαιτεί την παρατεταμένη παρουσία του στον χώρο στον οποίο βρίσκεται το θερμοκήπιο και του δίνει το πλεονέκτημα του απομακρυσμένου ελέγχου.

Αξίζει να αναφερθεί ωστόσο ότι το κόστος κατασκευής του είναι χαμηλό καθώς περιέχει εξαρτήματα μαζικής παραγωγής το οποίο τα καθιστά οικονομικά και συνεπώς η επεξεργαστική του ισχύς είναι χαμηλή με αποτέλεσμα να μπορεί να τρέξει ακόμα και Ultra low powered boards όπως το Raspberry pi zero που έχει τοποθετηθεί χωρίς αυτό να δημιουργεί πρόβλημα στην απόδοση και στην επίδοσή του. Με την χρήση των συγκεκριμένων εξαρτημάτων επίσης, του παρέχεται η δυνατότητα να έχει μεγάλη επεκτασιμότητα ως προς την ανάπτυξη του αλλά και να γίνει ευκολότερη η συντήρησή του καθώς και η επιδιόρθωση μελλοντικών βλαβών.

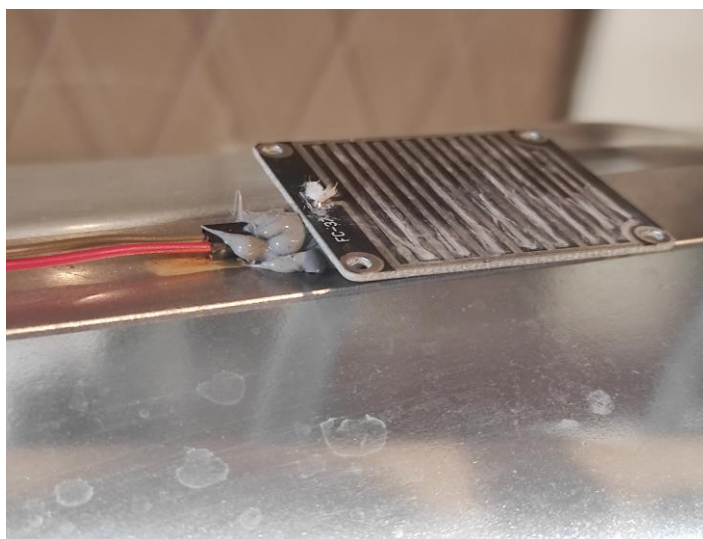
Τέλος, η σύνταξη του κώδικα έγινε με την χρήση της γλώσσας Python καθώς αυτή παρέχει ευκολία στον τρόπο γραφής του κώδικα, στην αποδοτικότερη μετανάστευση του και στην ταχύτερη εκτέλεση του στο πρόγραμμα. Το σημαντικότερο ωστόσο με την Python είναι ότι θα χρειάζονταν ελάχιστες μετατροπές του κώδικα ώστε να υλοποιηθεί σε πραγματικό μέγεθος το σύστημα μας.

Φυσικά το προτεινόμενο σύστημα μας δεν είναι τέλειο, αφού πάντα υπάρχει περιθώριο για βελτίωση καθώς οι λειτουργίες που καλύφθηκαν είναι οι πιο βασικές που χρειάζονται πράγματι σε ένα θερμοκήπιο.

Υπάρχουν και λειτουργίες που για διάφορους λόγους δεν υλοποιήθηκαν όπως είχαν σχεδιαστεί αρχικά ή απλά δεν έγιναν εφικτές για τεχνικούς λόγους. Ωστόσο, θα τις αναφέρουμε ως μελλοντικές προεκτάσεις.

1. Αυτόματος έλεγχος και αποθήκευση νερού.

Μια από αυτές είναι η συλλογή βρόχινου νερού με τη χρήση καναλιού το οποίο θα τίθεται σε λειτουργία όταν ένας αισθητήρας βροχής -συνδεδεμένος και ενεργός- θα λάμβανε σήμα ότι βρέχει. Τότε θα ενεργοποιούσε ένα εξωτερικό αισθητήρα που θα μετρούσε το Ph [16] του νερού. Στη συνέχεια ανάλογα με το αποτέλεσμα θα αποφάσιζε αν είναι υγιείς για τα φυτά ή όχι και θα έστελνε ανάλογο σήμα ώστε να ανοίξει θύρα του δοχείου συλλογής βρόχινου νερού για την αποθήκευση του.



Εικόνα 19 - Αισθητήρας Βροχής



Εικόνα 20 - Ph Sensor



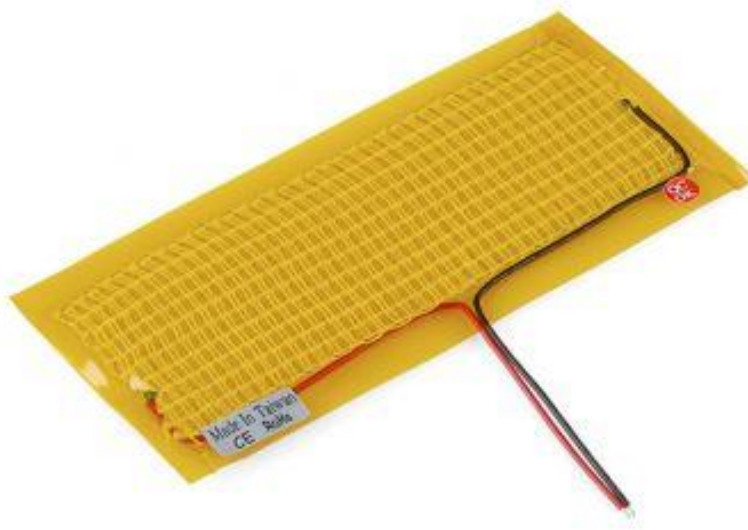
Εικόνα 21 - Λακκάκι για δρομολόγηση νερού προς αποθήκευση

2.Θερμαινόμενα Πατώματα

Έχοντας στην κατοχή μας αισθητήρες θερμοκρασίας για το έδαφος , μια καλή προσθήκη θα ήταν τα θερμαινόμενα πατώματα. Φυτεμένα στο έδαφος κάτω από τα ίδια τα φυτά σε μορφή λεπτού φύλλου φτιαγμένο από νήματα πολυεστέρα, στο εσωτερικό του οποίου υπάρχουν μικρές ίνες καλωδίων οι οποίες όταν συνδεθούν στο ρεύμα, θερμαίνονται. Το αποτέλεσμα είναι να θερμαίνουν το χώμα το χειμώνα που τα φυτά το έχουν περισσότερο ανάγκη.

Με τη χρήση των δεδομένων που παράγει ο αισθητήρας θερμοκρασίας χώματος, θα μπορούσαμε να προγραμματίσουμε στη βάση δεδομένων αποφάσεων μια ενέργεια υπεύθυνη για την ενεργοποίηση της θέρμανσης εδάφους όταν η θερμοκρασία που ανιχνεύεται από τους αισθητήρες λάβει τιμή χαμηλότερη από την ελάχιστη που όρισε ο γεωπόνος για το συγκεκριμένο

φυτό. Έτσι θα καλύπταμε την θερμότητα που δεν λαμβάνει το σύστημα από την ακτινοβολία του ηλίου με την αγωγή από τον εναλλάκτη.



Εικόνα 22

3.Τεχνητός Ήλιος

Μέσα από έρευνες που έχουν διεξάγει στο Πανεπιστήμιο του Όρεγκον στην Αμερική, αποδείχτηκε ότι τα φυτά χρησιμοποιούν το μπλε κομμάτι του ορατού φάσματος για την ανάπτυξη τους, φως δηλαδή που έχει μήκος κύματος μεταξύ 430 και 460 νανόμετρα. Το κόκκινο χρώμα του ορατού φάσματος κυμαίνεται μεταξύ των 650 και 700 νανομέτρων και ευθύνεται για την άνθηση και την ανάπτυξη της βλάστησης τους.

Έχοντας στην διάθεση μας την τεχνολογία LED , η οποία είναι μια οικονομική και εύκολα προσβάσιμη τεχνολογία, θα μπορούσαμε να εγκαταστήσουμε στα άκρα της οροφής του θερμοκηπίου μας ταινίες LED οι όποιες θα είχαν αυτά τα δυο χρώματα. Ακόλουθα, με τη χρήση αισθητήρα φωτός θα μπορούσαμε να δούμε πότε η ηλιακή ακτινοβολία κρίνεται ανεπαρκής και να προχωρήσουμε στο άνοιγμα του φωτός ανάλογα με την ανάγκη του φυτού μας.

Αξίζει να σημειωθεί ότι έγινε προσπάθεια με φως LED για την ενίσχυση της φωτοσύνθεσης μιας και το πειραματικό θερμοκήπιο μεταφέρθηκε σε εσωτερικό χώρο λόγω δυσμενών καιρικών συνθηκών. Δυστυχώς η κατασκευή πάσχει σε αυτόν τον τομέα, καθώς η έλλειψη αδιάβροχων μονώσεων στις θήκες των πλακετών αποτελεί μειονέκτημα.

Προκειμένου να “αντικατασταθεί” ο ήλιος η λάμπα LED θα έπρεπε να λειτουργεί για εννέα ώρες καθημερινά σε χρονικό ορίζοντα σχεδόν δυο εβδομάδων, παράγοντας τεχνητό φως ώστε τα φυτά να είναι υγιή και να αναπτύσσονται κανονικά. [17]



Εικόνα 23 - Red blue LED strip



Εικόνα 24 - Adafruit light Sensor



Εικόνα 25 - Τεχνικό φως με την χρήση ράβδου LED

- [1]<https://behrtech.com/blog/top-10-iot-sensor-types/>
- [2] <https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/>
- [3] <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1683-2018.pdf>
- [4]<https://www.eckerson.com/articles/data-management-best-practices-for-machine-learning>
- [5]<http://imu.ntua.gr/static/courses/strategicISmanagement/lectures/07-Knowledge%20Management.pdf>
- [6] https://repository.kallipos.gr/bitstream/11419/3382/1/02_chapter_04.pdf
- [7]<http://dl.n.jaipuria.ac.in:8080/jspui/bitstream/123456789/3111/1/The%20Internet%20of%20Things.pdf>
- [8]<http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>
- [9]https://www.sas.com/en_us/insights/big-data/what-is-big-data.html
- [10]<https://www.oracle.com/big-data/what-is-big-data.html>
- [11]https://en.wikipedia.org/wiki/Raspberry_Pi
- [12]https://en.wikipedia.org/wiki/Long_short-term_memory
- [13]<https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-error>
- [14]<https://keras.io/api/optimizers/adam/>
- [15]https://www.tensorflow.org/api_docs/python/tf/keras
- [16]<https://grobotronics.com/ph-sensor-kit.html>
- [17]<https://sciencing.com/led-lights-plant-growth-5958172.html>